

TUGAS AKHIR - TE 141599

**IMPLEMENTASI KONTROL PID PADA PERGERAKAN
AUTONOMOUS UNDERWATER VEHICLE DI DALAM
PIPA**

Yafet Yulio
NRP 2212100062

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Dr. Ir. Djoko Purwanto, M.Eng.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR - TE 141599

**IMPLEMENTASI KONTROL PID PADA PERGERAKAN
AUTONOMOUS UNDERWATER VEHICLE DI DALAM
PIPA**

Yafet Yulio
NRP 2212100062

Dosen Pembimbing
Dr. Muhammad Rivai, ST., MT.
Dr. Ir. Djoko Purwanto, M.Eng.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 141599

**IMPLEMENTATION OF PID CONTROL ON
AUTONOMOUS UNDERWATER VEHICLE MOVEMENT
INSIDE THE PIPE**

Yafet Yulio
NRP 2212100062

Supervisor
Dr. Muhammad Rivai, ST., MT.
Dr. Ir. Djoko Purwanto, M.Eng.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “Implementasi Kontrol PID pada Pergerakan Autonomous Underwater Vehicle di dalam Pipa” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2017



Yafet Yulio
22121000062

**IMPLEMENTASI KONTROL PID PADA
PERGERAKAN *AUTONOMOUS UNDERWATER
VEHICLE* DI DALAM PIPA**

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada**


**Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember**

Menyetujui:

Dosen Pembimbing I,

Dosen Pembimbing II,


Dr. Muhammad Rivai, ST., MT.
NIP. 196904261994031003


Dr. Ir. Djoko Purwanto, M.Eng.
NIP. 196512111990021002


**SURABAYA
JANUARI, 2017**

IMPLEMENTASI KONTROL PID PADA PERGERAKAN AUTONOMOUS UNDERWATER VEHICLE DI DALAM PIPA

Nama : Yafet Yulio
Pembimbing I : Dr. Muhammad Rivai, ST., MT.
Pembimbing II : Dr. Ir. Djoko Purwanto, M.Eng.

ABSTRAK

Jaringan pipa merupakan sarana transportasi penting bagi fluida seperti air, gas, dan minyak. Kebocoran pipa dapat mengakibatkan kerugian finansial yang besar bagi pihak pengguna sistem perpipaan. Penelitian ini bertujuan untuk membuat sistem robot yang dapat menyusuri saluran dalam pipa air untuk mendeteksi letak titik kebocoran agar kerugian finansial yang besar dapat dihindari. Penelitian ini menggunakan sensor jarak fotonik *infrared* yang dapat menjaga laju kestabilan *Autonomous Underwater Vehicle* (AUV) sehingga tidak terjadi tabrakan dengan dinding pipa. Komunikasi dengan sensor pada AUV menggunakan mikrokontroler Arduino Nano, sedangkan metode kontrol yang digunakan pada sistem ini adalah *Proportional-Integral-Derivative* (PID), yaitu untuk mengontrol kecepatan baling-baling AUV agar dapat menjaga laju kestabilan dalam pipa. Hasil penelitian menunjukkan bahwa AUV dapat bergerak di dalam pipa menggunakan kontrol PID untuk mengontrol pergerakan vertikal dan horisontal. Pengukuran sensor jarak di dalam air mengalami eror sebesar 1.25%. Untuk AUV bergerak secara vertikal, dengan rentang *duty cycle* sebesar 22% hingga 32% dan diperlukan K_p , K_i , K_d terbaik, masing-masing sebesar 3.5, 0, 0.1. Sedangkan untuk AUV bergerak horisontal, diperlukan K_p , K_i , K_d terbaik, masing-masing sebesar 2, 0, 0.1.

Kata-kata kunci : AUV, Kebocoran Pipa, PID.

Halaman ini sengaja dikosongkan

IMPLEMENTATION OF PID CONTROL ON AUTONOMOUS UNDERWATER VEHICLE MOVEMENT INSIDE THE PIPE

Name : Yafet Yulio
Supervisor : Dr. Muhammad Rivai, ST., MT.
Co-Supervisor : Dr. Ir. Djoko Purwanto, M.Eng

ABSTRACT

The pipeline is an important means of transportation for fluids such as water, gas, and oil. Leaking pipes can result in huge financial losses for the users of the piping system. This research aims to create a robot that can navigate the system lines in water pipes to detect the location of the leak points that huge financial losses can be avoided. This research uses distance sensor infrared photonic that can maintain the stability of the rate of Autonomous Underwater Vehicle (AUV) so there is no collision with the wall of the pipe. Communication with the AUV sensors use a microcontroller Arduino Nano, while the control method used in this system is a Proportional-Integral-Derivative (PID), which is to control the speed of the propeller AUV in order to keep the pace of stability in the pipeline. The results showed that the AUV can stir in the pipeline using PID control to control the vertical and horizontal movement. Measurement of the proximity sensor in the water in an error of 1.25%. For AUV move vertically, with a duty cycle range of 22% to 32% and required Kp, Ki, Kd best, respectively 3.5, 0, 0.1. As for the AUV to move horizontally, required Kp, Ki, Kd best, each at 2, 0, 0,1.

Keywords : AUV, Leakage Pipe, PID

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur atas rahmat yang diberikan oleh Tuhan Yang Maha Esa. Karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan penelitian ini. Selama pelaksanaan penelitian Tugas Akhir ini, penulis mendapatkan bantuan dari berbagai pihak baik dukungan secara moril maupun materiil. Terima kasih yang sebesar-besarnya penulis sampaikan kepada berbagai pihak yang mendukung dan membantu dalam tugas akhir ini, diantaranya :

1. Kedua orang tua tercinta, Prof. Dr. Budi Jatmiko, M.Pd. dan Dra. Sulistyorini yang tidak pernah putus untuk seluruh do'a, nasihat, motivasi, dan dukungannya.
2. Dr. Muhammad Rivai, ST., MT. selaku dosen pembimbing pertama, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
3. Dr.Ir. Djoko Purwanto, M.Eng, selaku dosen pembimbing kedua, atas bimbingan, inspirasi, pengarahan, dan motivasi yang diberikan selama pengerjaan penelitian tugas akhir ini.
4. Ir. Tasripan, M.T., Ir. Haris Pringadi, M.T., Dr.Ir. Hendra Kusuma, M.Eng.Sc., Astria Nur Irfansyah, S.T., M.Eng.PhD., selaku dosen penguji yang telah memberikan banyak masukan dalam penulisan tugas akhir.
5. Cicilia Anggi Mayangsari, Cornelia Gonstary, dan Okky Indra Prahasta yang tidak pernah putus untuk seluruh do'a, nasihat, motivasi, dan dukungannya.
6. Anas Maulidi Utama, Kamal Arief, dan Dheny Ashari yang selalu setia membantu dalam mengerjakan tugas akhir.
7. Teman-teman asisten Lab B202 yang senantiasa membantu dan memberikan semangat dalam mengerjakan tugas akhir.

Penulis sadar bahwa Tugas Akhir ini belum sempurna dan masih banyak hal yang dapat diperbaiki. Saran, kritik dan masukan baik dari semua pihak sangat membantu penulis untuk pengembangan lebih lanjut.

Surabaya, 05 Januari 2017



Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Penulisan.....	3
1.7 Relevansi	4
BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG	5
2.1 Aliran Fluida dalam Pipa	5
2.2 Hukum Archimedes	6
2.3 <i>Autonomous Underwater Vehicle</i>	7
2.4 Baterai Lithium Pollymer (Lipo).....	8
2.5 Sensor <i>Infrared</i>	9
2.6 <i>Pulse Width Modulation</i> (PWM).....	10
2.7 Mikrokontroler AVR ATmega 328	11
2.7.1 Pengertian AVR	11
2.7.2 Konfigurasi dan Arsitektur ATmega 328.....	13
2.8 Driver Motor	16
2.9 Sistem Pengendalian Motor DC pada Roda	16
2.10 Proportional-Integral-Derivative (PID).....	18
BAB III PERANCANGAN SISTEM.....	21
3.1 Perancangan Hardware.....	22
3.1.2 Rangkaian Penyuplai Tegangan	24
3.1.3 Sensor Jarak <i>Infrared</i> Sharp GP2Y0A41SKOF	25
3.1.4 <i>Driver</i> Motor	25
3.1.5 Motor DC	26
3.1.6 Arduino Nano.....	27
3.2 Perancangan Software	28
3.2.1 Kontrol PID.....	31

BAB IV PENGUJIAN DAN ANALISIS DATA	35
4.1 Pengujian Sensor <i>Infrared</i> Sharp GP2Y0A41SKOF	35
4.1.1 Pembuatan Sensor Agar Kedap Air	35
4.1.2 Pengujian di dalam Air	36
4.1.3 Penyesuaian Jarak di dalam Air	38
4.1.4 Hasil Pengujian Penyesuaian Jarak di dalam Air	39
4.2 Pengujian PWM pada Gerak Vertikal	39
4.3 Pengujian Kontrol PID pada Pergerakan AUV	40
4.3.1 Pergerakan Vertikal	40
4.3.1 Pergerakan Horisontal	42
4.3.2 Pergerakan di dalam Pipa	44
BAB V PENUTUP	45
5.1 Kesimpulan	45
5.2 Saran	45
DAFTAR PUSTAKA	47
LAMPIRAN	49
RIWAYAT HIDUP PENULIS	55

DAFTAR GAMBAR

Gambar 2.1 Eksperimen Ilustrasi Jenis Aliran	5
Gambar 2.2 Hukum Archimedes	6
Gambar 2.3 Sistem Koordinat AUV	7
Gambar 2.4 (a) Sonar mengirimkan sinyal ultrasonic, (b) <i>Receiver</i> menerima sinyal hasil pantulan	7
Gambar 2.5 Baterai Lipo	9
Gambar 2.6 (a) sensor fotonik mendeteksi objek, (b) sensor fotonik tidak mendeteksi objek	10
Gambar 2.7 Duty Cycle pada PWM	11
Gambar 2.8 AVR ATmega 328	12
Gambar 2.9 Arsitektur ATmega 328	13
Gambar 2.10 Konfigurasi Pin ATmega 328	14
Gambar 2.11 Skematik Sederhana <i>H-Bridge</i>	16
Gambar 2.12 Diagram blok pengendalian motor kanan pada metode <i>left wall following</i>	17
Gambar 3.1 Media Pengujian	21
Gambar 3.2 Blok Diagram Sistem	22
Gambar 3.3 AUV Tampak Depan	23
Gambar 3.4 AUV Memasuki Pipa	24
Gambar 3.5 Rangkaian Penyuplai Tegangan	24
Gambar 3.6 Sharp GP2Y0A41SKOF	25
Gambar 3.7 Driver Motor TB601A2	26
Gambar 3.8 Motor DC Pergerakan Horisontal	27
Gambar 3.9 Motor DC Pergerakan Vertikal	27
Gambar 3.10 Konfigurasi Arduino Nano	28
Gambar 3.11 Diagram Blok Pengarahan Posisi AUV	29
Gambar 3.12 <i>Flowchart</i> Perancangan <i>Software</i> AUV	30
Gambar 4.1 Pemberian Resin pada Sensor	35
Gambar 4.2 Uji Coba Sensor di dalam Air	36
Gambar 4.3 Grafik Jarak terhadap Tegangan	38
Gambar 4.4 AUV Mempertahankan Posisi Vertikal	41
Gambar 4.5 AUV Bergerak Maju Mempertahankan Posisi Horisontal	42
Gambar 4.6 AUV Melewati Pipa	44

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Perbedaan Sensor Infrared Thermal dan Sensor Infrared Photonic.....	14
Tabel 2.2 Konfigurasi Port B.....	14
Tabel 2.3 Konfigurasi Port C.....	15
Tabel 2.4 Konfigurasi Port D	15
Tabel 4.1 Hasil Pengujian Sensor di Air	36
Tabel 4.2 Hasil Pengujian Kalibrasi Sensor di Air	37
Tabel 4.3 Hasil Pengujian Sensor di Air Setelah Kalibrasi	39
Tabel 4.4 Hasil Pengujian PWM	40
Tabel 4.5 Hasil Tuning PID untuk Motor Pengerak Vertikal	41
Tabel 4.6 Hasil Tuning PID untuk Motor Pengerak Horizontal	43

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

1.1 Latar Belakang

Jaringan pipa merupakan sarana transportasi penting bagi fluida seperti air, gas, dan minyak. Kebocoran pipa dapat mengakibatkan kerugian finansial yang besar bagi pihak pengguna sistem perpipaan. Rata-rata kebocoran air Perusahaan Daerah Air Minum (PDAM) di Indonesia masih cukup tinggi yaitu 33% [1]. Banyak uang yang mengalir percuma dikarenakan PAM tidak dapat mengendalikan kebocoran air. Tingkat kebocoran tersebut dibedakan menjadi dua, yaitu faktor dalam dan faktor luar. Kebocoran pipa dikarenakan faktor dalam adalah kebocoran pipa dikarenakan umur pipa yang sudah tua dan harus diganti. Kebocoran pipa dikarenakan faktor luar adalah kebocoran pipa dikarenakan adanya pencurian air, sambungan liar, pembacaan meter yang tidak benar, akurasi meter yang rendah.

Secara umum, ada empat metode yang digunakan oleh PDAM untuk mendeteksi kebocoran pipa yaitu membandingkan pengukuran volume menggunakan *watermeter* dengan perhitungan volume menggunakan *flowmeter*, metode *sounding*, survei pelanggan, dan menggunakan alat berbasis *electroacoustic* [2]. Metode tersebut dirasa kurang cepat dan efisien dikarenakan saluran pipa yang ada di bawah tanah maupun di atas tanah sangat panjang sehingga sangat sulit dan membutuhkan waktu lama untuk mengetahui letak titik kebocoran. Dengan adanya alat yang dapat menyusuri saluran dalam pipa secara langsung, kemungkinan mengetahui letak titik kebocoran menjadi lebih cepat.

Penelitian tugas akhir ini dirancang untuk mengembangkan robot AUV agar dapat beroperasi di dalam pipa yang bergerak secara stabil dan tidak menabrak dinding dalam pipa.

1.2 Perumusan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini adalah:

1. Bagaimana AUV tidak menabrak dinding pipa?
2. Bagaimana menstabilkan kedalaman AUV dalam pipa?

1.3 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut:

1. Implementasi sensor jarak fotonik dengan cahaya *infrared* sebagai sensor untuk mengukur serta mengetahui letak dari dinding pipa yang berbeda jaraknya.
2. Implementasi kontrol PID untuk menjaga kestabilan AUV.

1.4 Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah:

1. Aliran fluida dalam pipa adalah aliran air laminar.
2. Jenis fluida yang digunakan berupa air.
3. Perbandingan besar diameter pipa dengan besar AUV adalah 3:1.

1.5 Metodologi

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut :

1. Studi Literatur

Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, dan artikel-artikel di internet dan forum-forum diskusi internet.

2. Perancangan *Hardware*

Perancangan *Hardware*, secara umum, meliputi 4 sensor jarak *infrared* dan 4 buah motor yang diberi baling-baling yang dikontrol dengan 1 buah Arduino Nano dan 2 buah Driver Motor di mana setiap driver motor mengontrol 2 buah motor. Dari keseluruhan tersebut akan dirancang menjadi sebuah kesatuan sistem elektronika yang terintegrasi.

3. Perancangan *Software*

Perancangan *Software*, meliputi perancangan algoritma kontrol PID yang digunakan pada pergerakan robot AUV agar dapat bergerak secara stabil dalam pipa air. Selain itu, pada

tahap ini dilakukan perancangan *software* untuk menghindari halangan berupa dinding pipa.

4. Pengujian Sistem

Pengujian alat dilakukan untuk menentukan keandalan dari sistem yang telah dirancang. Pengujian dilakukan untuk melihat apakah *software* dan *hardware* dapat bekerja secara baik.

Untuk pengujian dapat dilakukan dengan dua tahap. Pertama adalah menggunakan sensor jarak *infrared* menghadap bawah untuk menguji kemampuan robot AUV bergerak naik dan turun. Dan terakhir adalah menggunakan tiga buah sensor jarak *infrared* menghadap kanan, kiri dan depan untuk menguji kemampuan robot AUV bergerak maju dan berbelok ke kanan dan ke kiri.

5. Analisa

Analisa dilakukan terhadap hasil dari pengujian sehingga dapat ditentukan karakteristik dari *software* dan *hardware* yang telah dibuat. Apabila karakteristik dari *software* dan *hardware* masih belum sesuai, maka perlu dilakukan perancangan ulang pada sistem.

6. Penulisan Laporan Tugas Akhir

Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka dan teori penunjang, perancangan sistem, pengujian, dan penutup.

1.6 Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

➤ **BAB I : PENDAHULUAN**

Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

➤ **BAB II : TINJAUAN PUSTAKA DAN TEORI PENUNJANG**

Bab ini menjelaskan tentang teori penunjang dan *literature* yang dibutuhkan dalam pengerjaan tugas akhir ini. Dasar teori yang menunjang meliputi teori dasar sensor semikonduktor, Mikrokontroler Arduino Nano, kontrol PID, *pulse width modulation*, *driver motor*, motor DC. Bagian ini memaparkan mengenai beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Tugas Akhir ini.

➤ **BAB III : PERANCANGAN SISTEM**

Bab ini menjelaskan tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk sistem pengarahan posisi AUV secara otomatis.

➤ **BAB IV : PENGUJIAN**

Pada bab ini akan menjelaskan hasil uji coba sistem beserta analisisnya.

➤ **BAB V : PENUTUP**

Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangan lebih lanjut.

1.7 Relevansi

Hasil yang diharapkan dari tugas akhir ini diharapkan mampu meringankan pekerjaan manusia untuk melakukan inspeksi di dalam pipa air agar mudah mengetahui kondisi pipa. Pengembangan lebih lanjut dari sistem ini mampu beroperasi di dalam pipa dengan fluida yang berbeda, seperti minyak.

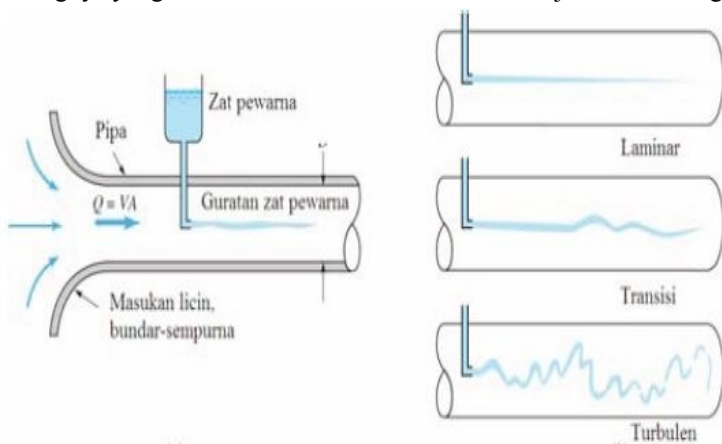
BAB II

TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Teori penunjang dalam bab ini menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem yang akan dibuat pada tugas akhir ini. Sedangkan tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan pernah diimplementasikan oleh penulis-penulis sebelumnya.

2.1 Aliran Fluida dalam Pipa

Aliran fluida di dalam sebuah pipa mungkin merupakan aliran laminar atau turbulen [3]. Seorang ahli matematika dan ilmuwan asal Inggris bernama **Osborne Reynolds** (1842-1912) pertama kali membedakan klasifikasi aliran ini menggunakan sebuah peralatan sederhana seperti gambar 2.1 di bawah ini. Pada saat air mengalir pada sebuah pipa dengan kecepatan rata-rata V dengan diameter D , sifat-sifat berikut dapat diamati dengan menyuntikkan zat pewarna yang mengambang seperti yang ditunjukkan. Untuk “laju aliran yang cukup kecil” guratan zat pewarna akan terlihat garis yang hanya sedikit menjadi kabur karena difusi molekuler dari zat pewarna dengan air di sekelilingnya yang dinamakan aliran laminar. Untuk “laju aliran sedang”,



Gambar 2.1 Eksperimen Ilustrasi Jenis Aliran [3]

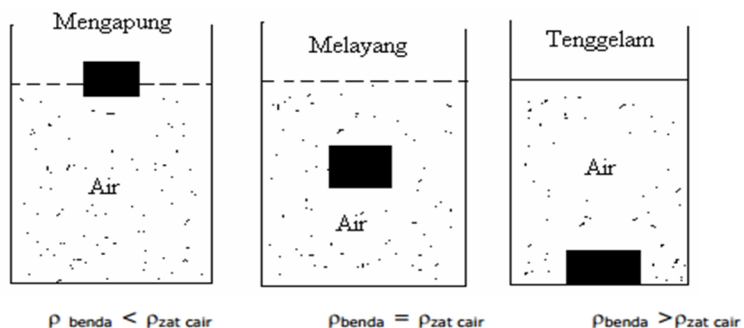
guratan zat pewarna berfluktuasi menurut waktu, ruang, dan olakan putus-putus dengan perilaku tak beraturan muncul di sepanjang guratan yang dinamakan aliran transisi. Untuk “laju aliran cukup besar” guratan zat pewarna dengan sangat segera menjadi kabur dan menyebar di seluruh pipa dengan pola acak yang dinamakan aliran turbulen.

2.2 Hukum Archimedes

Hukum Archimedes adalah sebuah hukum mengenai prinsip pengapungan di atas zat cair. Ketika sebuah benda tercelup sepenuhnya di dalam zat cair, maka akan ada gaya ke atas yang ditimbulkan oleh zat cair pada benda yang besarnya sama dengan berat zat cair yang dipindahkan. Hukum ini ditemukan oleh seorang ilmuwan yang bernama Archimedes seorang astronom, fisikawan, matematikawan dan juga insinyur berkebangsaan Yunani. Gambar 2.2 menunjukkan bagaimana hukum Archimedes bekerja pada saat benda memasuki air.

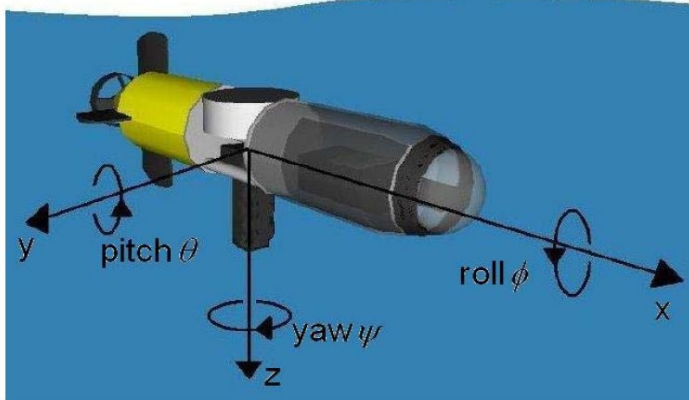
Berikut ini adalah syarat benda dikatakan mengapung, tenggelam, melayang :

- Mengapung : massa jenis benda harus lebih kecil dari masa jenis zat cair.
- Melayang : massa jenis benda harus sama dengan dari masa zat cair.
- Tenggelam : massa jenis benda harus lebih besar dari massa zat cair.



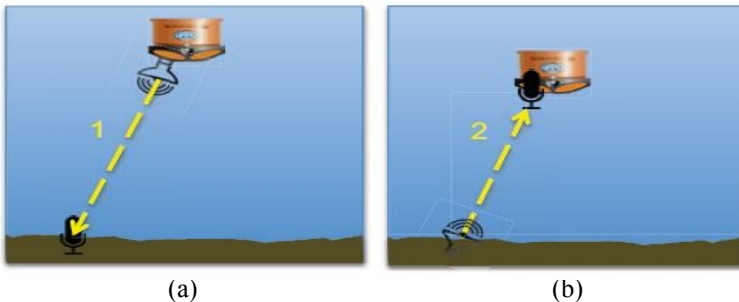
Gambar 2.2 Hukum Archimedes

2.3 *Autonomous Underwater Vehicle*



Gambar 2.3 Sistem Koordinat AUV [4]

AUV adalah kendaraan bawah air yang tak berawak dan dikendalikan oleh komputer. AUV dituntut dapat bermanuver tinggi dan sangat responsif karena beroperasi untuk menghindari rintangan dan pengenalan obyek dan lokasi [4]. Kebutuhan itu untuk mempertahankan jalur akurat pada ruang yang terbatas dan pengaruh aliran air agar dapat stabil. Gambar 2.3 menunjukkan AUV memiliki enam DOF, tiga koordinat khusus x, y, z dan sudut Euler yaw ψ pitch θ and roll ϕ .



Gambar 2.4 (a) Sonar mengirimkan sinyal ultrasonik, (b) *Receiver* menerima sinyal hasil pantulan [4]

AUV memanfaatkan sonar yang dapat mengetahui letak obyek yang ada di sekitarnya dengan cara mengirimkan sinyal ultrasonik seperti gambar 2.4(a). Sinyal yang dipantulkan oleh obyek akan dibaca oleh *receiver* pada AUV seperti gambar 2.4(b). Hal ini membuat AUV dapat menghindari obyek yang ada di depannya.

2.4 Baterai Lithium Pollymer (Lipo)

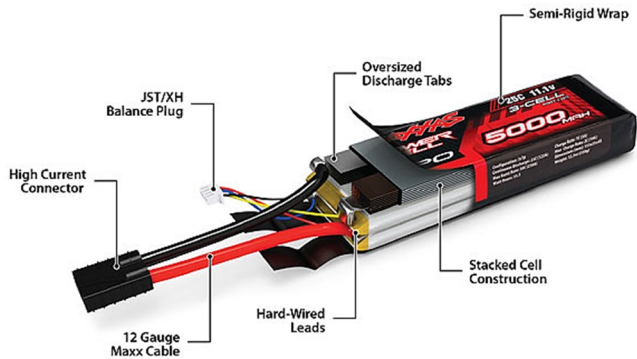
Baterai Lipo seperti pada gambar 2.5 digunakan sebagai sumber dalam suatu rangkaian elektronika. Baterai LiPo tidak menggunakan cairan sebagai elektrolit melainkan menggunakan elektrolit polimer kering yang berbentuk seperti lapisan plastik film tipis. Lapisan film ini disusun berlapis-lapis diantara anoda dan katoda yang mengakibatkan pertukaran ion. Dengan metode ini baterai LiPo dapat dibuat dalam berbagai bentuk dan ukuran. Diluar dari kelebihan arsitektur baterai LiPo, terdapat juga kekurangan yaitu lemahnya aliran pertukaran ion yang terjadi melalui elektrolit polimer kering. Hal ini menyebabkan penurunan pada charging dan discharging rate. Masalah ini sebenarnya bisa diatasi dengan memanaskan baterai sehingga menyebabkan pertukaran ion menjadi lebih cepat, namun metode ini dianggap tidak dapat untuk diaplikasikan pada keadaan sehari-hari. Seandainya para ilmuwan dapat memecahkan masalah ini maka risiko keamanan pada batera jenis lithium akan sangat berkurang.

Ada tiga kelebihan utama yang ditawarkan oleh baterai berjenis LiPo daripada baterai jenis lain seperti NiCad atau NiMH yaitu :

- Baterai LiPo memiliki bobot yang ringan dan tersedia dalam berbagai macam bentuk dan ukuran
- Baterai LiPo memiliki kapasitas penyimpanan energi listrik yang besar
- Baterai LiPo memiliki tingkat discharge rate energi yang tinggi

Selain keuntungan yang dimilikinya, baterai jenis ini juga memiliki beberapa kelemahan yaitu:

- Harga baterai LiPo masih tergolong mahal jika dibandingkan dengan baterai jenis NiCad dan NiMH
- Performa yang tinggi dari baterai LiPo harus dibayar dengan umur yang lebih pendek. Usia baterai LiPo sekitar 300-400 kali siklus 31 pengisian ulang. Sesuai dengan perlakuan yang diberikan pada beterei.



Gambar 2.5 Baterai Lipo

- Alasan keamanan. Baterai LiPo menggunakan bahan elektrolit yang mudah terbakar.
- Baterai LiPo membutuhkan penanganan khusus agar dapat bertahan lama. Charging, Discharging, maupun penyimpanan dapat mempengaruhi usia dari baterai jenis ini.

2.5 Sensor *Infrared*

Sensor *infrared* adalah sensor yang bereaksi pada radiasi sinar *infrared* (IR). Sinar *infrared* memiliki panjang gelombang berkisar 700nm-1mm yang tidak tampak oleh mata manusia. Sensor *infrared* dibedakan menjadi 2 tipe kerja seperti gambar 2.1, yaitu :

1. Sensor *Infrared Thermal*

Sensor *Infrared Thermal* memanfaatkan radiasi *infrared* yang diradiasikan oleh objek. Sensor *Infrared Thermal* pada umumnya disebut sensor *infrared* pasif karena hanya terdiri dari *receiver* dari bahan semikonduktor yang peka terhadap radiasi emisi sinar *infrared*, sehingga cara kerjanya hanya menerima emisi sinar *infrared* yang pada umumnya dipancarkan oleh objek akibat suhu yang dimilikinya.

2. Sensor *Infrared Photonic*

Sensor *Infrared Photonic* terdiri dari pemancar dan penerima sinar *infrared* tergabung menjadi sebuah perangkat. Penerima

menerima pantulan sinar *infrared* yang dipancarkan oleh pemancar, oleh karena itu pada umumnya sensor *infrared photonic* disebut sensor *infrared* aktif.

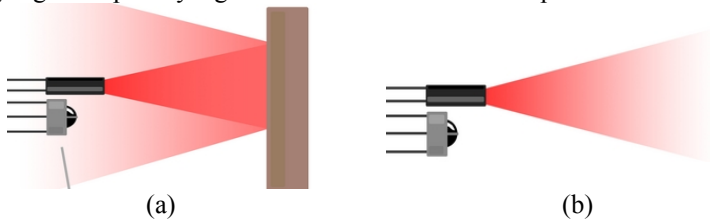
Tabel 2.1 Perbedaan Sensor *Infrared Thermal* dan Sensor *Infrared Photonic*

Sensor <i>Infrared Thermal</i>	Sensor <i>Infrared Photonic</i>
Tidak memiliki perangkat pemancar sendiri	Memiliki perangkat pemancar sendiri
Memiliki permasalahan pengukuran 2 parameter	Mampu secara langsung mengukur jarak dan kecepatan objek
Tidak berfungsi di kegelapan dan objek yang bersuhu dingin	Mampu berfungsi secara efektif di terang, gelap, dan objek yang bersuhu dingin

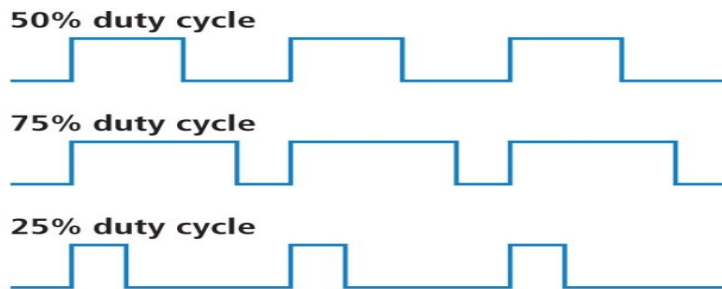
Pada gambar 2.6 (a) menunjukkan pemancar sensor fotonik memancarkan sinar *infrared* dan mengenai objek, hasil pantulan sinar *infrared* tersebut diterima oleh penerima pada sensor fotonik sehingga sensor tersebut memberikan sinyal bahwa di depan terdapat objek. Pada gambar 2.6 (b) menunjukkan penerima sensor fotonik tidak dapat menerima sinar *infrared* yang dipancarkan oleh pemancar sensor karena tidak ada objek yang memantulkan sinar *infrared* tersebut.

2.6 *Pulse Width Modulation (PWM)*

Dasar pulse width modulation (PWM) secara luas digunakan di dalam aplikasi elektronika daya untuk pengaturan pengkonversian daya (DC/DC, DC/AC, dll.) [5]. Secara sederhana, PWM merupakan sinyal yang lebar pulsa yang bernilai “HIGH” dalam satu periode



Gambar 2.6 (a) sensor fotonik mendeteksi objek, (b) sensor fotonik tidak mendeteksi objek



Gambar 2.7 Duty Cycle pada PWM

mewakili suatu tegangan DC, tergantung nilai duty cycle. Duty cycle merupakan perbandingan lama waktu sinyal bernilai “HIGH” dengan satu periode. Gambar 2.7 menggambarkan duty cycle dari PWM. Selain digunakan dalam pengaturan pengkonversian daya, ada beberapa aplikasi lain dari PWM. Contoh aplikasi umum yang lain adalah pengendalian kecepatan motor DC, pengendalian motor servo, pengaturan nyala terang LED dan lain sebagainya.

2.7 Mikrokontroler AVR ATMega 328

2.7.1 Pengertian AVR

Nama AVR sendiri berasal dari "Alf (Egil Bogen) and Vegard (Wollan) 's Riscprocessor" dimana Alf Egil Bogen dan Vegard Wollan adalah dua penemu berkebangsaan Norwegia yang menemukan mikrokontroler AVR yang kemudian diproduksi oleh atmel.

Mikrokontroler adalah piranti elektronik berupa IC (Integrated Circuit) yang memiliki kemampuan manipulasi data (informasi) berdasarkan suatu urutan instruksi (program). Dalam sebuah struktur mikrokontroler akan kita temukan juga komponen-komponen seperti: processor, memory, clock, dll.

Salah satu arsitektur mikrokontroler yang terdapat di pasaran adalah jenis AVR (Advanced Virtual RISC). Arsitektur mikrokontroler jenis AVR ini pertama kali dikembangkan pada tahun 1996 oleh dua orang mahasiswa Norwegian Institute of Technology yaitu Alf-Egil Bogen dan Vegard Wollan. Dalam perkembangannya, AVR dibagi menjadi beberapa



Gambar 2.8 AVR ATmega 328 [6]

varian yaitu AT90Sxx, ATmega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing varian adalah kapasitas memori dan beberapa fitur tambahan saja.

Pemrograman mikrokontroler AVR dapat menggunakan low level language (assembly) dan high level language (C, Basic, Pascal, JAVA, dll) tergantung compiler yang digunakan. Salah satu yang banyak dijumpai di pasaran adalah AVR tipe ATmega, yang terdiri dari beberapa versi, yaitu: ATmega8535, ATmega16, ATmega328, ATmega32, ATmega324P, ATmega644, ATmega644P dan ATmega128.

Gambar 2.8 adalah AVR ATmega 328 yang memiliki beberapa fitur antara lain :

- 32 x 8- bit register serba guna.
- Memiliki SRAM (*Static Random Acces Memory*) sebesar 2 kb.
- 130 macam instruksi yang hampir semuanya dieksekusi dalam satu siklus *clock*.
- 32 kb *Flash Memory* dan pada arduino memiliki *bootloader* yang menggunakan 2 kb dari *flash memory* sebagai *bootloader*.
- Kecepatan mencapai 16 MIPS dengan *clock* 16 MHz.
- Memiliki EEPROM (*Electrically Erasable Programmable Read Only Memory*) sebesar 1 kb sebagai tempat penyimpanan data semi permanen karena EEPROM tetap dapat menyimpan data meskipun catu daya dimatikan.
- *Master / Slave SPI Serial interface*.

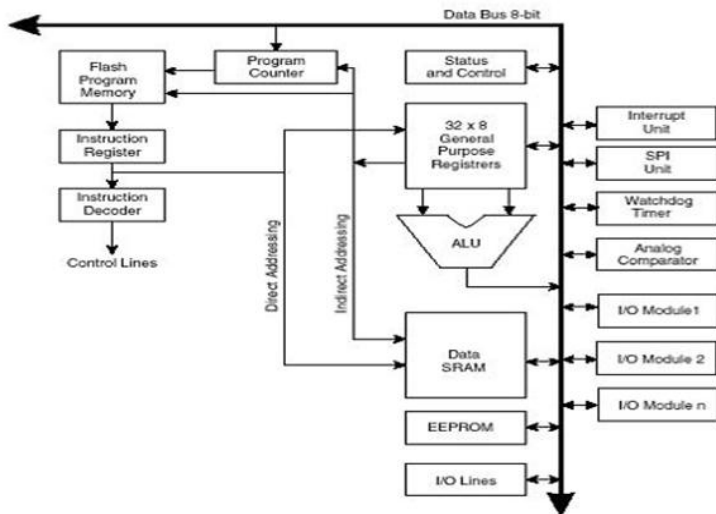
- Memiliki pin I/O digital sebanyak 14 pin dan 6 diantaranya PWM (*Pulse Width Modulation*) output.

2.7.2 Konfigurasi dan Arsitektur ATmega 328

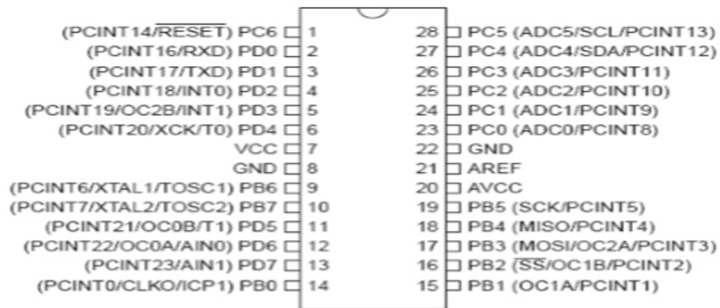
ATmega328 mempunyai arsitektur RISC (Reduce Instruction Set Computer) yang dimana setiap proses eksekusi data lebih cepat dari pada arsitektur CISC (Completed Instruction Set Computer) seperti gambar 2.9 [6].

Mikrokontroller ATmega 328 memiliki arsitektur Harvard, dimana memori untuk kode program dan memori untuk data dipisahkan sehingga dapat memaksimalkan kerja dan parallelism. Instruksi – instruksi dalam memori program dieksekusi dalam satu alur tunggal, dimana pada saat satu instruksi dikerjakan instruksi berikutnya sudah diambil dari memori program. Konsep inilah yang memungkinkan instruksi – instruksi dapat dieksekusi dalam setiap satu siklus clock.

32 x 8-bit register serba guna digunakan untuk mendukung operasi pada ALU (Arithmetic Logic Unit) yang dapat dilakukan dalam satu siklus. 6 dari register serbaguna ini dapat digunakan sebagai 3 buah register pointer 16-bit pada mode pengalamatan tak langsung untuk



Gambar 2.9 Arsitektur ATmega 328 [6]



Gambar 2.10 Konfigurasi Pin ATmega 328

mengambil data pada ruang memori data. Ketiga register pointer 16-bit ini disebut dengan register “X” (gabungan R26 dan R27), register Y (gabungan R28 dan R29), dan register Z (gabungan R30 dan R31). Hampir semua instruksi AVR memiliki format 16-bit. Setiap alamat memori program terdiri dari instruksi 16-bit atau 32-bit. Selain register serba guna di atas, terdapat register lain yang terpetakan dengan teknik memory mapped I/O selebar 64 byte. Beberapa register ini digunakan untuk fungsi khusus antara lain sebagai register control Timer/ Counter, Interupsi, ADC, USART, 17 SPI, EEPROM, dan ungsi I/O lainnya. Register – register ini menempati memori pada alamat 0x20h – 0x5Fh.

Tabel 2.2 Konfigurasi Port B

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2) PCINT7 (Pin Change Interrupt 7)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1) PCINT6 (Pin Change Interrupt 6)
PB5	SCK (SPI Bus Master clock Input) PCINT5 (Pin Change Interrupt 5)
PB4	MISO (SPI Bus Master Input/Slave Output) PCINT4 (Pin Change Interrupt 4)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2A (Timer/Counter2 Output Compare Match A Output) PCINT3 (Pin Change Interrupt 3)
PB2	SS (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output) PCINT2 (Pin Change Interrupt 2)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output) PCINT1 (Pin Change Interrupt 1)
PB0	ICP1 (Timer/Counter1 Input Capture Input) CLKO (Divided System Clock Output) PCINT0 (Pin Change Interrupt 0)

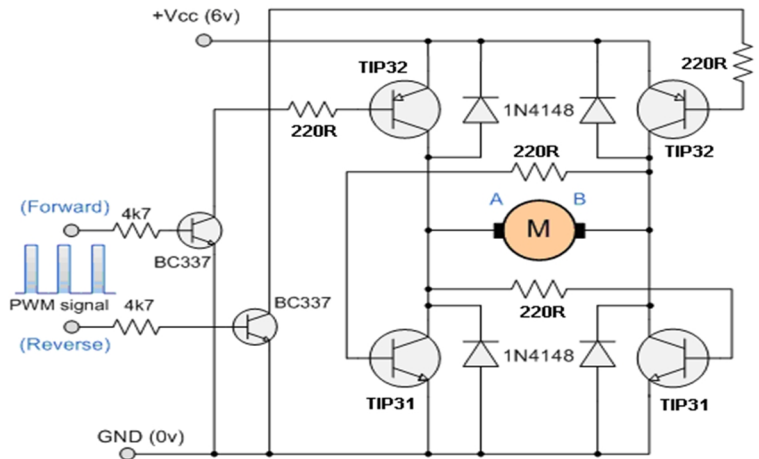
Tabel 2.3 Konfigurasi Port C

Port Pin	Alternate Function
PC6	RESET (Reset pin) PCINT14 (Pin Change Interrupt 14)
PC5	ADC5 (ADC Input Channel 5) SCL (2-wire Serial Bus Clock Line) PCINT13 (Pin Change Interrupt 13)
PC4	ADC4 (ADC Input Channel 4) SDA (2-wire Serial Bus Data Input/Output Line) PCINT12 (Pin Change Interrupt 12)
PC3	ADC3 (ADC Input Channel 3) PCINT11 (Pin Change Interrupt 11)
PC2	ADC2 (ADC Input Channel 2) PCINT10 (Pin Change Interrupt 10)
PC1	ADC1 (ADC Input Channel 1) PCINT9 (Pin Change Interrupt 9)
PC0	ADC0 (ADC Input Channel 0) PCINT8 (Pin Change Interrupt 8)

Tabel 2.4 Konfigurasi Port D

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input) PCINT23 (Pin Change Interrupt 23)
PD6	AIN0 (Analog Comparator Positive Input) OC0A (Timer/Counter0 Output Compare Match A Output) PCINT22 (Pin Change Interrupt 22)
PD5	T1 (Timer/Counter 1 External Counter Input) OC0B (Timer/Counter0 Output Compare Match B Output) PCINT21 (Pin Change Interrupt 21)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input) PCINT20 (Pin Change Interrupt 20)
PD3	INT1 (External Interrupt 1 Input) OC2B (Timer/Counter2 Output Compare Match B Output) PCINT19 (Pin Change Interrupt 19)
PD2	INT0 (External Interrupt 0 Input) PCINT18 (Pin Change Interrupt 18)
PD1	TXD (USART Output Pin) PCINT17 (Pin Change Interrupt 17)
PD0	RXD (USART Input Pin) PCINT16 (Pin Change Interrupt 16)

2.8 Driver Motor



Gambar 2.11 Skematik *H-Bridge* [7]

Driver motor merupakan rangkaian yang memperkuat arus yang akan masuk ke motor dari pengontrol atau mikrokontroler sehingga arus yang masuk ke motor cukup besar untuk menggerakkan motor. Rangkaian motor yang sering dipakai adalah rangkaian H-Bridge.

Rangkaian H-Bridge namanya berasal dari rangkaian full-bridge yang ditunjukkan oleh gambar 2.11 [7]. Dengan rangkaian ini, arah gerak motor bisa diatur. Jika saklar A dan saklar D disambungkan dan saklar B dan saklar C tidak disambungkan, arus akan mengalir dari tegangan sumber ke sisi kiri motor. Hal ini akan membuat gerakan motor searah jarum jam. Sebaliknya, jika saklar yang tersambung adalah saklar B dan saklar C, arus akan mengalir dari tegangan sumber ke sisi kanan motor. Hal ini akan membuat motor bergerak berlawanan arah jarum jam.

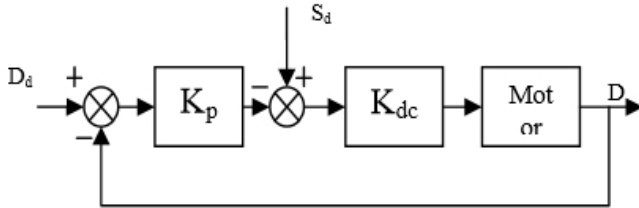
2.9 Sistem Pengendalian Motor DC pada Roda

Pengendalian motor DC bertujuan untuk menentukan seberapa cepat robot akan berjalan, bagaimana teknik pengaturan ketika belok kanan, belok kiri, memutar dan sebagainya [8]. Yang terpenting dalam pengendalian motor DC disini adalah bagaimana caranya robot tidak

menabrak dinding ataupun halangan yang ada disekitarnya cara yang dapat digunakan agar robot tidak menabrak dinding kiri atau kanan yaitu dengan tetap menjaga jarak antara robot dengan dinding kiri atau kanan pada nilai tertentu. Pada metoda ini robot menggunakan sensor jarak kiri untuk mendeteksi jarak robot ke dinding kiri dan sensor jarak kanan untuk mendeteksi jarak robot ke dinding kanan. Metoda yang dapat digunakan yaitu *left wall following* (mengikuti dinding kiri) dan *right wall following* (mengikuti dinding kanan).

- Left Wall Following

Diagram pengontrolan motor kanan pada metoda left wall following ditunjukkan pada gambar 2.12. Diagram pengontrolan untuk motor kiri adalah sama dengan gambar 2.12 kecuali bahwa output K_{pw} dijumlahkan dengan S_d .



Ket :

S_d : Kecepatan yang diinginkan

D_d : Jarak yang diinginkan antara robot dengan dinding kiri

K_{pw} : Konstanta proporsional yang menyatakan seberapa cepat sistem menanggapi error jarak

D : Jarak antara robot dengan dinding kiri

K_{dc} : Konstanta duty cycle

Gambar 2.12 Diagram blok pengendalian motor kanan pada metode *left wall following* [8]

Dari gambar 2.12 diperoleh persamaan untuk duty cycle yang diberikan ke motor kanan (DC_R) dan motor kiri (DC_L) sebagai berikut:

$$DC_R = S_d \cdot K_{dc} - (D_d - D) K_{pw} \cdot K_{dc} \quad (2.1)$$

$$DC_L = S_d \cdot K_{dc} + (D_d - D) K_{pw} \cdot K_{dc} \quad (2.2)$$

Dari persamaan tersebut, pada saat $D < D_d$ maka motor kiri akan dipercepat dan motor kanan diperlambat sehingga motor akan menjauhi dinding kiri sedangkan apabila $D > D_d$ maka motor kiri akan diperlambat dan motor kanan dipercepat sehingga robot berjalan mendekati dinding kiri. Setelah $D = D_d$ maka robot akan berjalan konstan dengan kecepatan S_d .

- Right Wall Following

Diagram pengontrolan motor kanan dan kiri dan kiri pada metoda ini sama dengan diagram pengontrolan pada metoda *left wall following*. Yang membedakan adalah nilai D , disini D merupakan jarak robot terhadap dinding kanan. Dari gambar tersebut diperoleh persamaan untuk mencari duty cycle untuk motor kanan (DC_R) dan motor kiri (DC_L) :

$$DC_R = S_d.K_{dc} + (D_d - D).K_{pw}.K_{dc} \quad (2.3)$$

$$DC_L = S_d.K_{dc} - (D_d - D).K_{pw}.K_{dc} \quad (2.4)$$

Dari persamaan di atas terlihat apabila jarak robot terlalu ke kanan ($D < D_d$) maka motor kanan akan dipercepat sedangkan motor kiri diperlambat sehingga robot akan berjalan menjauhi dinding kanan dan sebaliknya jika $D > D_d$ maka motor kanan akan diperlambat sedangkan motor kiri dipercepat sehingga robot akan berjalan mendekati dinding. Setelah $D = D_d$ maka robot akan berjalan lurus konstan dengan kecepatan S_d . Sedangkan pengontrolan Motor DC terhadap dinding yang ada di depannya dilakukan dengan cara mengecek jarak robot terhadap dinding depannya. selama jarak robot dengan dinding depan masih dalam posisi aman maka kontrol motor menggunakan metoda *left wall following* (mengikuti dinding kiri) dan *right wall following* (mengikuti dinding kanan). Dan apabila jarak berada pada jarak minimum maka robot akan mengurangi besarnya Duty cycle yang diberikan pada motor atau melakukan tindakan lain seperti berhenti, belok kanan atau kiri.

2.10 Proportional-Integral-Derivative (PID)

Sistem kontrol PID merupakan suatu kontroller yang berfungsi untuk menentukan kepresisian (kestabilan) suatu system instrumentasi dengan karakteristik umpan balik pada sistem tersebut [9]. Kontroler PID terdiri

dari tiga parameter yaitu *proportional* (P), *derivative* (D), *integral* (I). Dalam implementasinya masing-masing parameter dapat bekerja sendiri maupun menggabungkan dari parameter tersebut. Parameter P, I dan D merupakan parameter yang diatur dalam sistem sesuai terhadap input sistem yang diinginkan.

Dasar teoritis untuk menganalisis kinerja kontrol PID adalah mempertimbangkan representasi dari *Integrator* dengan *Transformasi Laplace* (1/S), dan *Differentiator* dengan *Transformasi Laplace* (S). Secara konseptual, kontroler PID memiliki tiga representasi yang berbeda, yaitu Pertama, ada representasi simbolis, dimana masing-masing dari tiga parameter dapat dipilih untuk mencapai tindakan kontrol yang berbeda. Kedua, ada bentuk operator domain waktu. Ketiga, ada *Transformasi Laplace* versi kontroler PID. Hal ini memberikan interpretasi kontrol s- domain dan memungkinkan hubungan antara domain waktu dan domain frekuensi.

Berikut merupakan pola algoritma kontroler PID yang terdiri dari elemen *proportional gain*, *integral gain*, dan *derivative gain*. Dalam pola algoritma kontroler PID juga terdapat *error* yang berfungsi sebagai pengontrolan output saat ini dengan output.

Sebelumnya :

$$P = K_p e(t) + K_i \int e(t) dt + K_d \frac{d}{dt} e(t) \quad (2.5)$$

Diumpamakan $PID = U_n$

$$U_n = K_p e_n + K_i \sum e_n n_i = 0 + K_d (e_n - e_{n-1}) \quad (2.6)$$

Keterangan (Persamaan PID) :

Dimana variable k_p, k_i, k_d merupakan gain dari parameter proportional, integral, derivative dan $e_n - e_{n-1}$ adalah nilai error saat ini dan sebelumnya. Dalam persamaan algoritma PID diperlukan time sampling (dt) untuk waktu pengambilan data, sehingga output kontroler PID (U_n) dapat diamati per sampling data yang telah ditentukan. Serta pada algoritma PID terdapat istilah integral yang diartikan sebagai bentuk penjumlahan secara increments, dan differensial yang diartikan sebagai bentuk perbedaan atau selisih.

Parameter-Parameter Penyusun Kontroller PID :

- **Proportional**

Pengaruh pada sistem :

1. Menambah atau mengurangi kestabilan.
2. Mengurangi (bukan menghilangkan) Error steady state

Persamaan matematis :

$$U(t) = K_p \cdot e(t) \quad (2.7)$$

Dimana K_p : konstanta proporsional

Dalam laplace

$$\frac{U(s)}{E(s)} = K_p \quad (2.8)$$

- **Integral**

Pengaruh pada sistem :

Menghilangkan *Error Steady State* namun dapat menimbulkan ketidakstabilan (karena menambah orde sistem). Selain itu kontrol integral memiliki rise time yang lebih lama dibandingkan kontrol proporsional (responnya lebih lama).

Persamaan matematis :

$$U(t) = K_i \int_0^t e(t) dt \quad (2.9)$$

Dimana K_i : konstanta integral

Dalam laplace :

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \quad (2.10)$$

- **Derivative**

Pengaruh pada sistem :

Memberikan efek redaman pada osilasi. Hal ini juga dapat menyebabkan memberikan nilai K_p yang lebih besar. Selain itu kontrol *Derivative* hanya berubah saat perubahan error, saat ada error statis kontroler ini tidak dapat bereaksi, sehingga kontroler D ini tidak dapat digunakan sendiri tanpa ada kontroler lainnya (biasanya di gabungkan dengan pengendali proporsional). Memperbaiki respon transien, karena memberikan aksi saat ada perubahan error.

Persamaan matematis :

$$D = K_d \frac{d}{dt} e(t) \quad (2.11)$$

BAB III

PERANCANGAN SISTEM

Perancangan sistem ini meliputi perangkat keras dan perangkat lunak. Perangkat keras yang digunakan antara lain, Arduino Nano, *driver* motor TB601A2, motor dc. Sensor yang digunakan merupakan sensor jarak *infrared* Sharp GP2Y0A41SKOF. Sensor ini merupakan sensor yang dapat mengidentifikasi jarak dengan menggunakan cahaya *infrared*.

Perangkat lunak meliputi program Arduino Nano menggunakan *software* Arduino dan kontrol PID. Kontrol PID adalah sistem kontrol yang digunakan untuk mengontrol kestabilan AUV. Bab ini menjelaskan secara keseluruhan dan desain tiap-tiap blok yang ada. Perangkat sistem ini didesain dengan sumber baterai lipo 3 sel 1100 mah 11.1 volt untuk *driver* motor dan 5 volt untuk Arduino Nano.

Media pengujian pada penelitian ini berupa sebuah pipa seperti gambar 3.1 dengan diameter 30cm dan panjang 90cm. Pipa tersebut diletakkan terendam sepenuhnya di dalam sebuah akuarium dengan ukuran panjang 120cm, lebar 40cm, dan tinggi 45cm.

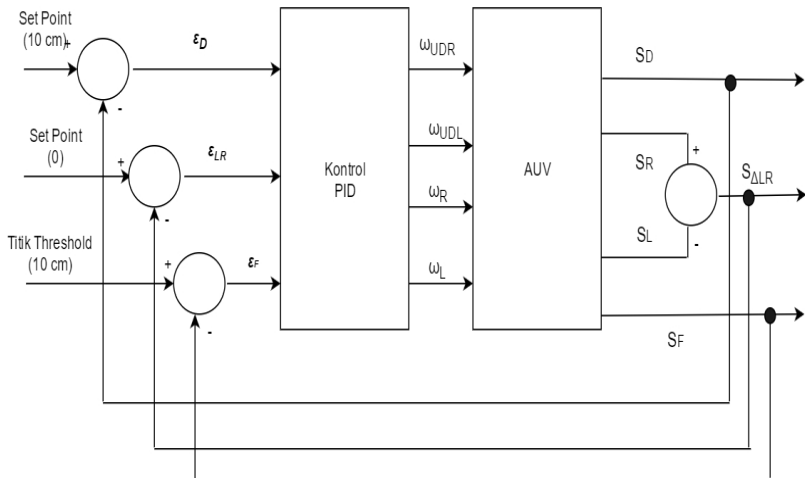


Gambar 3.1 Media Pengujian

3.1 Perancangan Hardware

Sistem yang diajukan untuk penelitian Tugas akhir ini menggunakan mikrokontroler Arduino Nano sebagai unit pengolahan data dan pengeksekusi kontrol PID. Secara garis besar, cara kerja sistem terlihat pada gambar 3.2. Pertama, untuk kondisi bergerak naik-turun sistem diberi masukan berupa *set point* sebesar 10, dan kanan-kiri diberi *set point* sebesar 0. Setelah itu, *set point* tersebut dikurangi dengan *feedback* yang berasal dari sensor bawah dan selisih sensor kanan-kiri sehingga menghasilkan error (ε_D dan ε_L). Error tersebut dikelolah di dalam sistem kontrol sehingga menghasilkan kecepatan putaran pada motor DC (ω_{UDR} dan ω_{UDL} untuk motor dc bergerak vertikal, ω_R dan ω_L untuk motor dc bergerak horisontal) dan membuat AUV berada di posisi tengah dalam keadaan melayang.

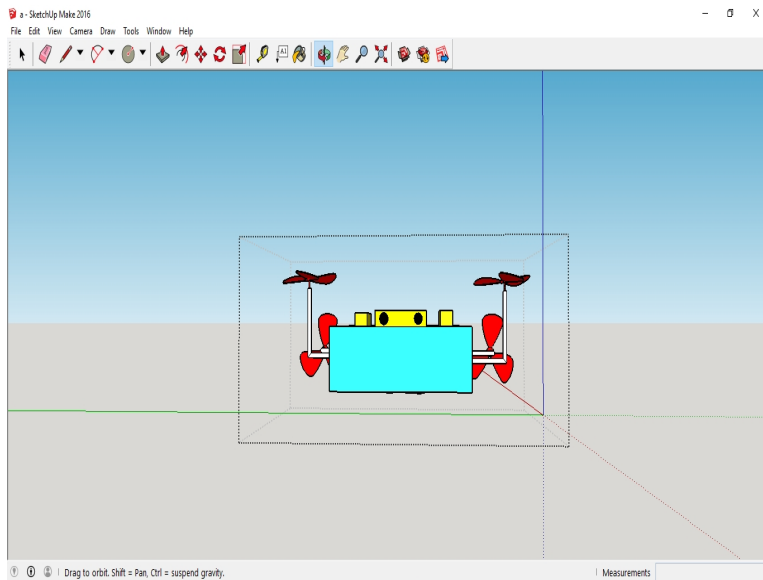
Untuk kondisi maju, sistem diberi masukan berupa titik *threshod* sebesar 10 cm. Setelah itu, titik *threshod* tersebut dikurangi dengan *feedback* yang berasal dari sensor depan sehingga menghasilkan error (ε_F). Jika jarak yang terukur dari sensor depan kurang dari titik *threshod* yang ditentukan, maka AUV akan bergerak maju. Namun, jika kurang dari titik *threshod* tersebut, AUV akan berhenti dan hanya dikontrol kanan-kiri dan atas-bawah.



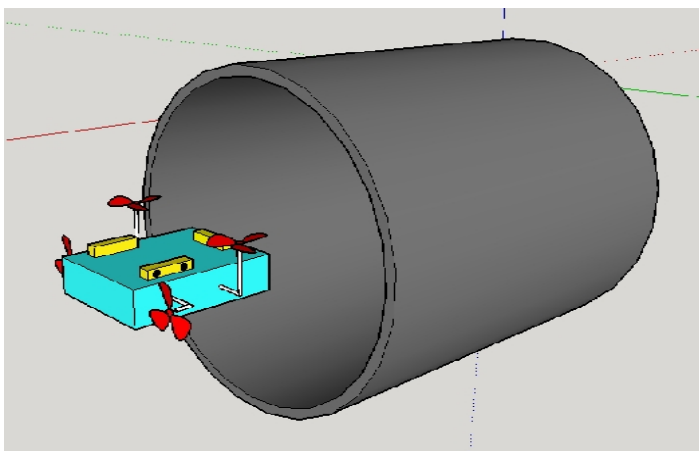
Gambar 3.2 Blok Diagram Sistem

Sistem AUV terdiri empat buah sensor jarak *infrared* yang diletakkan di atas sebanyak tiga sensor (untuk depan, kanan dan kiri) dan di bawah sebanyak satu sensor (untuk bawah) pada sebuah kotak yang berukuran (12 cm x 9 cm x 6 cm) yang kedap udara agar pada saat dimasukkan air, kotak tidak dapat kemasukan air. Dapat dilihat pada gambar 3.3 dan gambar 3.4 merupakan ilustrasi sistem yang dibuat. Sistem AUV ini digerakkan oleh empat buah motor DC yang diatur 2 *driver* motor, dimana dua buah motor DC untuk naik dan turun, dan dua buah motor DC untuk maju.

Dalam perencanaannya, sistem ini seharusnya dapat mempertahankan posisinya pada saat berada di air dan dapat melewati pipa dengan baik tanpa mengenai dinding pipa. Saat memasuki pipa, sensor jarak *infrared* akan membaca jarak antara AUV dengan dinding pipa. Hasil dari yang dibaca oleh sensor tersebut akan dikelololah oleh mikrokontroler sehingga membuat AUV mempertahankan posisinya di tengah.

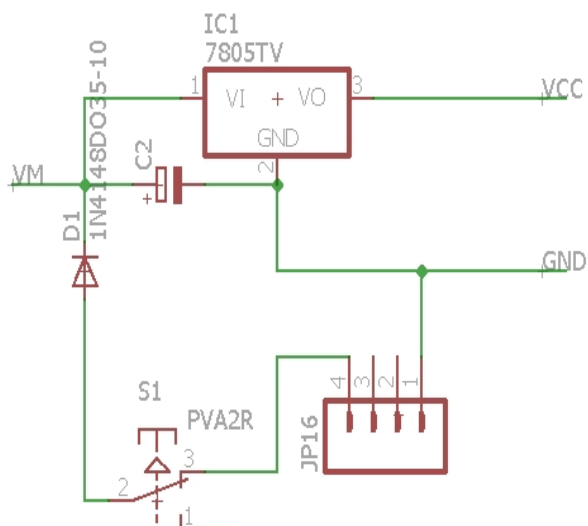


Gambar 3.3 AUV Tampak Depan



Gambar 3.4 AUV Memasuki Pipa

3.1.2 Rangkaian Penyuplai Tegangan



Gambar 3.5 Rangkaian Penyuplai Tegangan

Pada tugas akhir ini digunakan rangkaian penyuplai tegangan yang digunakan untuk menyuplai mikrokontroler Arduino Nano, *driver* motor, dan motor DC. Pada perancangan *power supply* untuk menyuplai motor dc dengan tegangan output 11.1 volt dan *supply* mikrokontroler menggunakan tegangan output 5 volt terdapat pada gambar 3.5.

Macam-macam komponen yang dibutuhkan yaitu : satu buah Baterai Lipo 3 sel 1100 mah dan satu buah IC7805 dengan arus output 1 ampere.

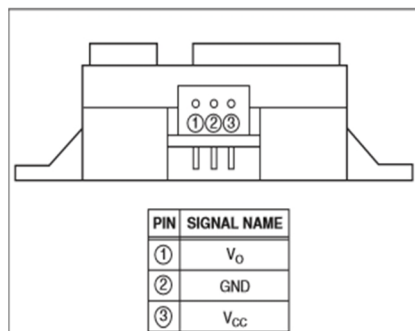
3.1.3 Sensor Jarak *Infrared* Sharp GP2Y0A41SKOF

Pada tugas akhir ini digunakan empat buah sensor jarak *infrared* Sharp GP2Y0A41SKOF. Sensor ini merupakan sensor untuk mengukur jarak, yang terdiri dari kombinasi terintegrasi dari PSD (*Position Sensitive Detector*), IR-LED (*Infrared Emitting Diode*) dan rangkaian pemrosesan sinyal.

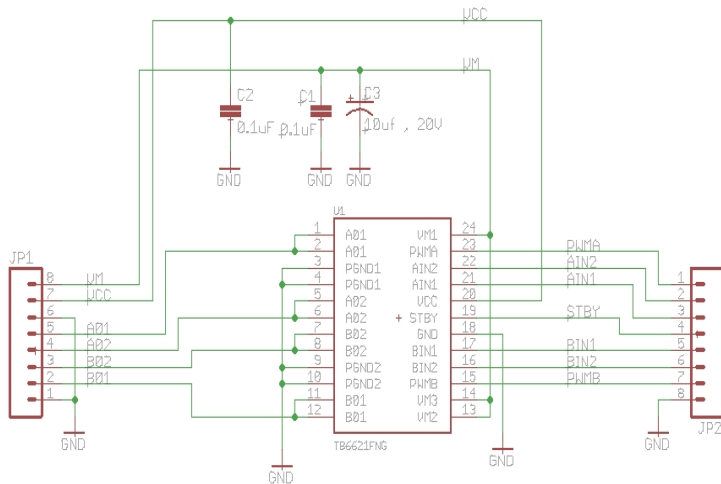
Pada gambar 3.6, untuk mengoperasikan sensor ini diperlukan *supply* tegangan 4.5-5.5 Volt pada pin 3, *ground* pada pin 2. Tegangan akan keluar melalui pin 1. Jarak minimal yang dapat terdeteksi adalah 4 cm dan memiliki jarak maksimal 30 cm.

3.1.4 *Driver* Motor

Driver motor yang digunakan pada tugas akhir ini adalah TB601A2. Pada gambar 3.7, *driver* motor ini terdapat 2 buah pin PWM sehingga AUV membutuhkan sebanyak 2 buah. Satu *driver* untuk menggerakkan 2 buah motor bergerak horisontal dan *driver* lainnya untuk menggerakkan 2 buah motor bergerak vertikal.



Gambar 3.6 Sharp GP2Y0A41SKOF



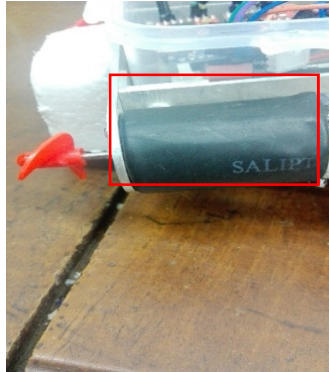
Gambar 3.7 Driver Motor TB601A2

Driver motor yang digunakan pada tugas akhir ini adalah TB601A2. Pada gambar 3.7, *driver* motor ini terdapat 2 buah pin PWM sehingga AUV membutuhkan sebanyak 2 buah. Satu *driver* untuk menggerakkan 2 buah motor bergerak horisontal dan *driver* lainnya untuk menggerakkan 2 buah motor bergerak vertikal.

Tegangan maksimal dari *driver* motor ini sebesar 15 Volt. Satu buah *driver* motor dapat mengontrol 2 buah pwm, dimana masing2 pwm digunakan untuk mengontrol satu buah motor. *Driver* ini membutuhkan tegangan input sebesar 2.7-5.5 Volt.

3.1.5 Motor DC

Pergerakan vertikal dan horisontal sistem AUV menggunakan motor DC dengan tegangan 12 Volt. Arah pergerakan motor ditentukan oleh pemberian tegangan DC pada kedua input motor. Gambar 3.8 dan gambar 3.9 memberikan gambaran motor DC yang digunakan pada tugas akhir ini beserta peletakkannya.



Gambar 3.8 Motor DC Pergerakan Horisontal

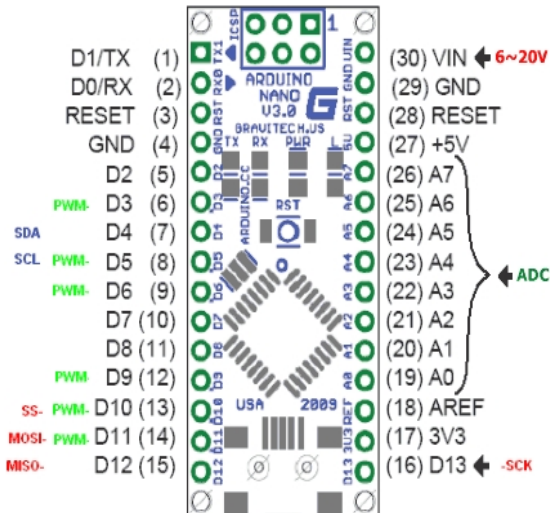
3.1.6 Arduino Nano

Pada tugas akhir ini digunakan Mikrokontroler Arduino Nano untuk mengontrol sistem AUV dengan kontrol PID agar AUV berjalan stabil. Dibutuhkan tegangan 5 Volt untuk dapat mengoperasikan mikrokontroler ini. Pada sistem ini dibutuhkan 4 input analog untuk membaca 4 buah sensor yaitu pada pin A.2 hingga pin A.5 dan 4 output PWM untuk mengontrol 2 buah *driver* motor yaitu pada pin A.3, A.9, A.10, A.11.



Gambar 3.9 Motor DC Pergerakan Vertikal

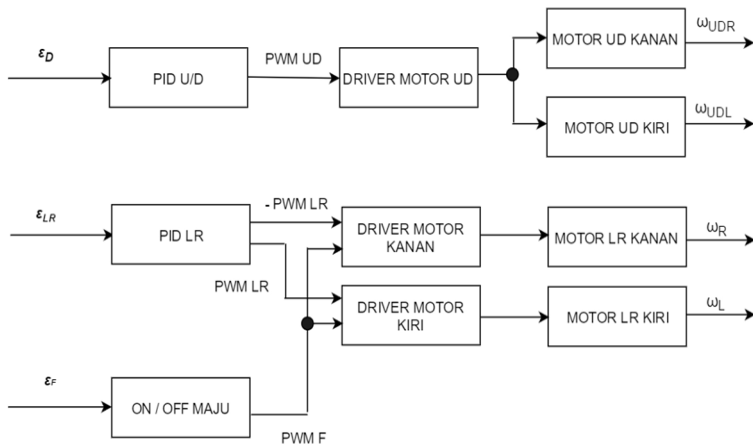
Berikut gambar 3.10 adalah konfigurasi pin pada Arduino Nano yang digunakan :



Gambar 3.10 Konfigutasi Arduino Nano

3.2 Perancangan Software

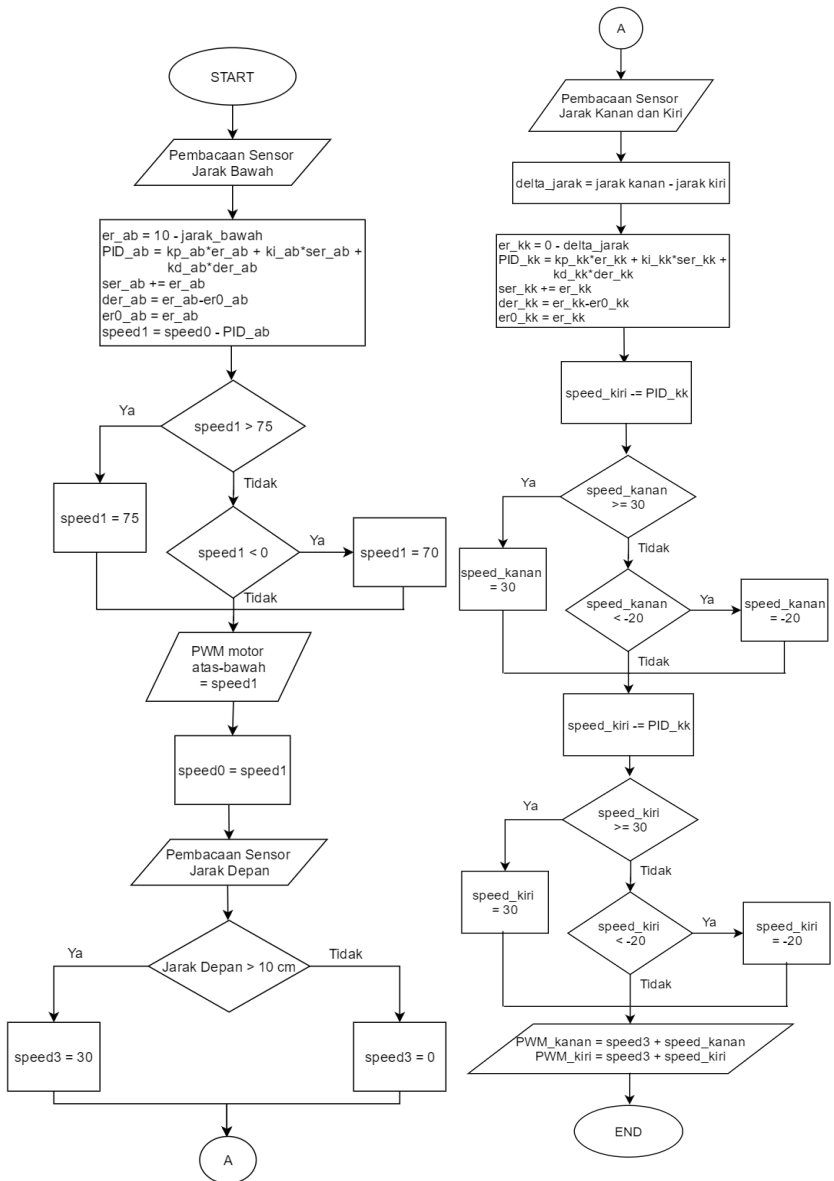
Pada bagian ini, semua proses dilakukan di dalam Mikrokontroler Arduino Nano. Error dari pembacaan sensor jarak bawah (ε_D) digunakan sebagai kontrol input PID. Hasil output dari kontrol PID berupa PWM sebagai input bagi motor *driver*. Motor *driver* yang dipakai akan menghasilkan arus untuk menggerakkan motor dc pergerakan vertikal kanan dan kiri. Error dari pembacaan sensor jarak kanan dan kiri (ε_L) digunakan sebagai kontrol input PID. Hasil output dari kontrol PID berupa PWM sebagai input bagi motor *driver*. Motor *driver* yang dipakai akan menghasilkan arus untuk menggerakkan motor dc pergerakan horisontal kanan dan kiri. Error dari pembacaan sensor jarak depan (ε_F) digunakan sebagai kontrol input on/off. Hasil output dari kontrol on/off berupa PWM sebagai input bagi motor *driver*. Motor *driver* yang dipakai akan menghasilkan arus untuk menggerakkan motor dc pergerakan horisontal kanan dan kiri. Diagram blok dari sistem ada pada Gambar 3.11.



Gambar 3.11 Diagram Blok Pengarahan Posisi AUV

Pada gambar 3.12 merupakan *flowchart* perancangan *software* AUV. Mula-mula sensor jarak bawah melakukan pembacaan lalu diproses di dalam kontrol PID. Nilai er_ab merupakan eror hasil sensor jarak bawah yang berasal dari *set point* (10 cm) dikurangi dengan jarak bawah yang terbaca. Nilai ser_ab merupakan sigma eror atas-bawah hasil dari penjumlahan sigma eror atas-bawah dan eror atas-bawah. Nilai der_ab merupakan derivative atas-bawah hasil dari eror atas-bawah dikurangi eror atas-bawah sebelumnya ($er0_ab$). Hasil perhitungan itu dimasukkan dalam persamaan kontrol PID sehingga PID atas-bawah (PID_ab) merupakan hasil dari penjumlahan kontanta proporsional (kp_ab) dikali eror atas-bawah dijumlah kontanta integral (ki_ab) dikali dengan sigma eror atas-bawah dijumlah konstanta derifatif (kd_ab) dikali derivative atas-bawah. Hasil persamaan tersebut mempengaruhi PWM motor untuk menggerakkan AUV secara vertikal. PWM motor atas-bawah ($speed1$) besarnya menjadi kecepatan awal ($speed0$) dikurangi hasil PID atas-bawah.

Pada tahap kedua sensor jarak depan melakukan pembacaan lalu diproses di dalam kontrol on/off. Dengan pemberian titik *threshold* 10 cm, jika pembacaan sensor depan masih melebihi 10 cm, maka kecepatan awal motor penggerak horisontal ($speed2$) sebesar 30, sedangkan jika sebaliknya, maka kecepatan awal motor penggerak horizontal sebesar 0.



Gambar 3.12 Flowchart Perancangan Software AUV

Pada tahap ketiga sensor jarak kanan-kiri melakukan pembacaan lalu diproses di dalam kontrol PID. Nilai er_kk merupakan eror hasil sensor jarak kanan-kiri yang berasal dari *set point* (0 cm) dikurangi dengan delta jarak kanan dengan jarak kiri. Nilai ser_kk merupakan sigma eror kanan-kiri hasil dari penjumlahan sigma eror kanan-kiri dan eror kanan-kiri. Nilai der_kk merupakan derivative kanan-kiri hasil dari eror kanan-kiri dikurangi eror kanan-kiri sebelumnya ($er0_kk$). Hasil perhitungan itu dimasukkan dalam persamaan kontrol PID sehingga PID kanan-kiri (PID_kk) merupakan hasil dari penjumlahan kontanta proporsional (kp_kk) dikali eror kanan-kiri dijumlah kontanta integral (ki_kk) dikali dengan sigma eror kanan-kiri dijumlah konstanta derifatif (kd_kk) dikali derivative kanan-kiri. Hasil persamaan tersebut mempengaruhi PWM motor untuk menggerakkan AUV secara horisontal. PWM motor kanan besarnya menjadi kecepatan awal ($speed2$) ditambah kecepatan motor kanan ditambah PID kanan-kiri, sedangkan PWM motor kiri besarnya menjadi kecepatan awal ($speed2$) ditambah kecepatan motor kiri dikurangi PID kanan-kiri.

3.2.1 Kontrol PID

Keluaran kontrol PID dalam Tugas Akhir ini adalah sinyal PWM yang akan mengontrol jumlah arus yang dihasilkan *driver* motor untuk menggerakkan motor dengan kecepatan yang diinginkan. Perlu dilakukan *tuning* K_p , K_i , dan K_d secara manual, nilai dari masing-masing koefisien tersebut akan membuat respon motor cepat dan meminimumkan timbulnya osilasi.

Implementasi kontrol PID di dalam Tugas Akhir ini berada di Mikrokontroler Arduino Nano. Algoritma yang digunakan untuk mengontrol kecepatan motor untuk bergerak maju dan naik-turun berbeda. Perbedaanannya pada tipe kontrol PID, dimana motor untuk kanan-kiri dikontrol untuk *differential* sedangkan untuk atas bawah tidak.

Algoitma dari kontrol PID yang dirancang untuk sistem ini meliputi pencarian nilai sinyal *error*, dilanjutkan dengan pencarian *integral error* dan nilai *derivative error*. Setelah ketiga nilai didapatkan, perumusan kontrol PID akan memperoleh nilai keluaran PWM. Algoritma ini diimplementasikan di Arduino Nano dengan *syntax* seperti berikut ini.

- Mencari *proportional error*
 - Bergerak naik-turun :

- er_ab = SETPOINT_BAWAH - jarak_bawah;
 - Bergerak maju-kanan-kiri :
er_kk = SETPOINT_KANAN_KIRI - delta_jarak;
- Mencari *integral error*
 - Bergerak naik-turun :
ser_ab += er_ab;
if (ser_ab > 65535 && ser_ab < -65535) ser_ab = 0;
 - Bergerak maju-kanan-kiri :
ser_kk += er_kk;
if (ser_kk > 65535 && ser_kk < -65535) ser_kk = 0;

Syntax di atas ini bertujuan agar jika integral error mengalami overflow, nilai integral error dinolkan kembali.
- Mencari *derivative error*
 - Bergerak naik-turun :
der_ab = er_ab - er0_ab;
er0_ab = er_ab;
 - Bergerak maju-kanan-kiri :
der_kk = er_kk - er0_kk;
er0_kk = er_kk;
- Keluaran kontrol PID
 - Bergerak naik-turun :
speed1 = speed0 - PID_ab;
if (speed1 > 180) speed1 = 180;
speed0 = speed1;
 - Bergerak maju-kanan-kiri :
if (er_kk > 1.0) speed_kanan = PID_kk;
else speed_kanan = 0;
 - if (er_kk < -1.0) speed_kiri = -1 * PID_kk;
else speed_kiri = 0;
- Keluaran PWM
 - if (nMotor >= 1 && nMotor <= 4)
 - {
 - switch (nMotor)
 - {
 - case 1:
 - motorA = dir_a;
 - pwm = pwm_a;
 - break;

```

case 2:
    motorA = dir_b;
    pwm    = pwm_b;
    break;
case 3:
    motorA = dir_c;
    pwm    = pwm_c;
    break;
case 4:
    motorA = dir_d;
    pwm    = pwm_d;
    break;
default:
    break;
}

```

Syntax di atas merupakan perintah untuk menentukan motor mana yang akan menyala setelah diberi pwm.

```

switch (command)
{
case FORWARD:
    digitalWrite (motorA, HIGH);
    analogWrite(pwm,speed);
    break;
case BACKWARD:
    digitalWrite (motorA, LOW);
    analogWrite(pwm,speed);
    break;
case BRAKE:
    digitalWrite (motorA, LOW);
    analogWrite(pwm,0);
case RELEASE:
    digitalWrite (motorA, LOW);
    analogWrite(pwm,0);
default:
    break;
}
}

```

Syntax di atas merupakan perintah untuk menentukan arah putaran motor agar maju, mundur, atau diam.

Halaman ini sengaja dikosongkan

BAB IV

PENGUJIAN DAN ANALISIS DATA

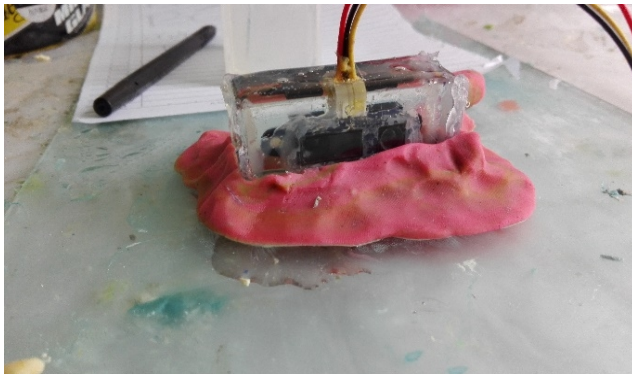
Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Bab ini bertujuan untuk mengetahui apakah tujuan dalam perancangan sistem pada tugas akhir ini telah terlaksana atau tidak. Pengujian pada bab ini terdiri dari pengujian sensor *infrared* Sharp GP2Y0A41SKOF pengujian PWM pada Gerak Vertikal, dan pengujian kontrol PID pada pergerakan AUV.

4.1 Pengujian Sensor *Infrared* Sharp GP2Y0A41SKOF

Pengujian sensor *infrared* dilakukan dengan 3 tahap, pertama diujikan di dalam, setelah itu melakukan penyesuaian jarak di dalam air, dan diujikan kembali di air.

4.1.1 Pembuatan Sensor Agar Kedap Air

Agar sensor tidak terjadi arus pendek saat dipakai di dalam air, maka perlu dilakukan pengedapan untuk sensor tersebut. Gambar 4.1 menunjukkan pembuatan sensor agar kedap air. Untuk membuat sensor kedap air, sensor ditemplei kaca tipis agar cahaya sensor masih bisa tembus. Setelah itu, bagian sensor yang tidak tertutupi oleh kaca diberi larutan resin yang digabung dengan catalys sehingga sensor tersebut akan aman saat terkena air.



Gambar 4.1 Pemberian Resin pada Sensor

4.1.2 Pengujian di dalam Air



Gambar 4.2 Uji Coba Sensor di dalam Air

Pada tahap ini dilakukan uji coba pengukuran sensor yang sudah dibuat kedap di dalam air. Pada gambar 4.2, dapat dilihat sensor masih bisa membaca jarak walaupun berada di dalam air dikarenakan sensor tersebut memancarkan sinar *infrared* sehingga medium air masih bisa ditembus oleh sinar tersebut.

Dari pengujian tersebut didapatkan hasil sebagai berikut :

Tabel 4.1 Hasil Pengujian Sensor di Air

Jarak Sebenarnya (cm)	Jarak di Air (cm)
5	3.63
6	4.28
7	4.83
8	5.3
9	5.99
10	6.52
11	7.06
12	7.81

Tabel 4.2 Hasil Pengujian Kalibrasi Sensor di Air

Jarak Sebenarnya (cm)	Pembacaan Tegangan (volt)
4	1.9
5	1.78
6	1.68
7	1.55
8	1.45
9	1.39
10	1.3
11	1.24
12	1.2
13	1.14
14	1.09
15	1.07
16	1.05
17	1.01
18	0.99
19	0.96
20	0.94
21	0.97
22	0.99
23	0.96

Dari data yang diperoleh dari Tabel 4.1, eror yang terlihat mencapai 33.21%. Hal ini dikarenakan perbedaan udara dan air memiliki kerapatan optik yang berbeda sehingga arah berkas cahaya yang diteruskan berbeda.

4.1.3 Penyesuaian Jarak di dalam Air

Sensor *infrared* Sharp GP2Y0A41SKOF memiliki karakteristik sendiri jika digunakan di dalam air. Oleh karena itu dilakukan kalibrasi untuk sensor ini agar dapat digunakan di dalam air. Dari pengujian kalibrasi sensor di dalam air didapatkan data sebagai berikut :

Dari data yang terdapat pada Tabel 4.2, dapat diubah menjadi grafik jarak terhadap tegangan sehingga dapat terlihat kurva yang dihasilkan oleh hasil pengujian kalibrasi sensor tersebut.

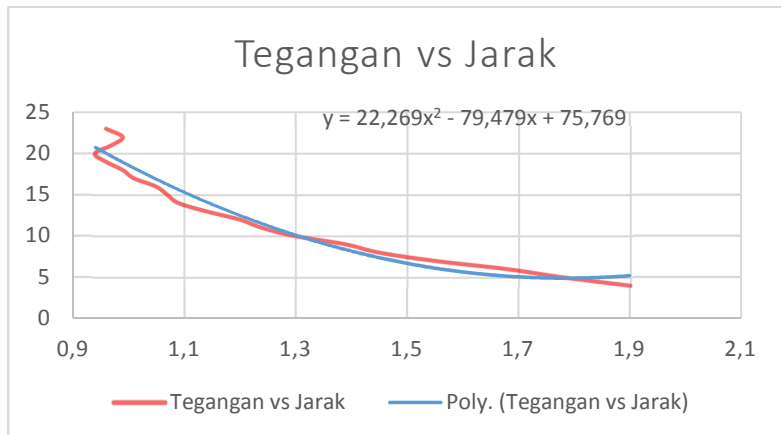
Dari Gambar 4.3 di atas, garis berwarna merah merupakan data hasil pengukuran kalibrasi sensor yang dilakukan di air dan garis berwarna biru merupakan garis polynomial orde dua yang mendekati garis persamaan data hasil pengukuran kalibrasi sensor tersebut. Sehingga dari perbandingan kedua garis tersebut didapatkan persamaan karakteristik baru sensor Sharp GP2Y0A41SKOF yang dapat membaca jarak yang ada di dalam air dengan lebih akurat. Persamaannya adalah :

$$y = 22.269x^2 - 79.479x + 75.769 \quad (4.1)$$

Dimana :

y = Jarak yang terukur di dalam air

x = Tegangan yang terukur di dalam air



Gambar 4.3 Grafik Jarak terhadap Tegangan

4.1.4 Hasil Pengujian Penyesuaian Jarak di dalam Air

Taberl 4.3 Hasil Pengujian Sensor di Air Setelah Kalibrasi

Jarak Sebenarnya	Jarak Setelah dikali
5	5.31
6	6.16
7	7.14
8	8.06
9	9.05
10	10.03
11	11.03
12	12.07

Pada tahap ini dilakukan pengujian sensor setelah dilakukan kalibrasi sensor di air dengan persamaan baru (persamaan 4.1). Dari pengujian tersebut didapatkan data sebagai berikut :

Dari data yang diperoleh dari Tabel 4.3, eror yang terlihat mencapai 1.25%. Hasil tersebut menunjukkan masih adanya eror yang muncul dikarenakan faktor kemiringan sensor saat dilakukan pengujian sehingga sedikit menimbulkan jarak bertambah sedikit.

4.2 Pengujian PWM pada Gerak Vertikal

Pengujian pwm pada gerak vertikal bertujuan agar dapat mengetahui kemampuan minimum hingga maksimum motor DC gerak vertikal dapat mendorong robot AUV sampe ke dasar. Dari pengujian ini didapatkan data sebagai berikut ;

Tabel 4.4 Hasil Pengujian PWM

Duty Cycle (%)	Penurunan (cm)
21.57	0
23.53	1
25.49	2
27.45	3
29.41	5
31.37	7

Dari Tabel 4.4, dapat dilihat *duty cycle* 21.57% tidak dapat membuat motor DC untuk mendorong AUV ke bawah, Sebaliknya dengan *duty cycle* 31,37% dapat membuat motor DC untuk mendorong AUV sampai ke dasar (penurunan maksimum).

4.3 Pengujian Kontrol PID pada Pergerakan AUV

Pada tahapan ini, AUV diberi kontrol PID agar dapat beroperasi secara otomatis sehingga dapat menghindari halangan dengan baik tanpa menabraknya sama sekali. Dilakukan 3 tahap pengujian, yaitu pengujian AUV pada pergerakan vertikal, pengujian AUV pada pergerakan horizontal, dan yang terakhir pengujian pada pergerakan AUV di dalam pipa.

4.3.1 Pergerakan Vertikal

Pada pengujian ini, Nilai Kp, Ki, dan Kd diubah-ubah dan dicari agar pergerakan motor AUV beresilasi seminimum mungkin ketika diberi suatu *setpoint*



Gambar 4.4 AUV Mempertahankan Posisi Vertikal

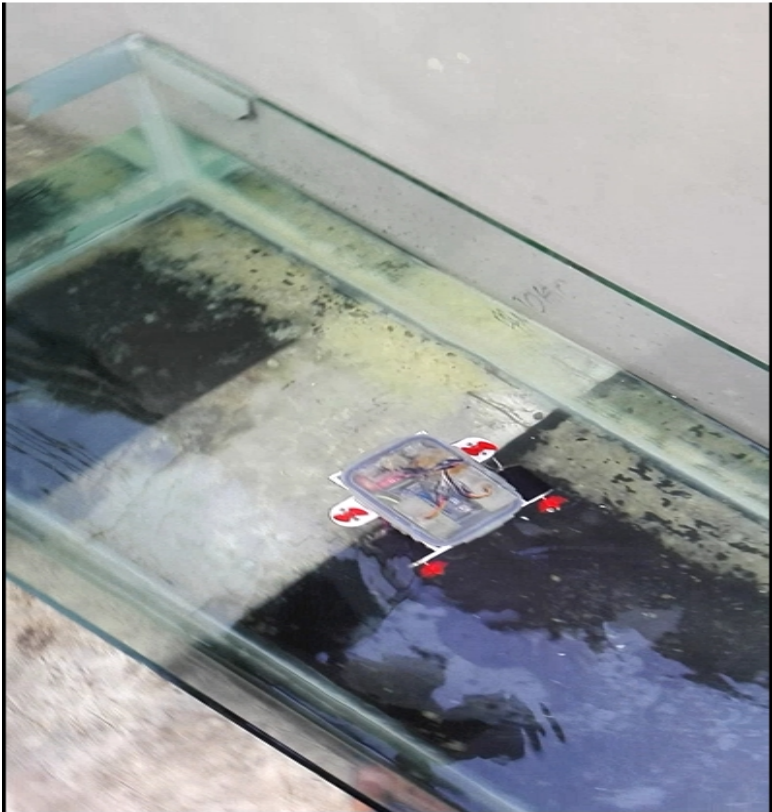
Gambar 4.4 merupakan peristiwa di saat AUV mempertahankan posisi vertikal dengan bergerak naik-turun. Hasil pengujian yang didapatkan sebagai berikut:

Tabel 4.5 Hasil Tuning PID untuk Motor Penggerak Vertikal

No	Kp	Ki	Kd	Duty Cycle Maksimum	Simpangan Osilasi	Lama Osilasi
1	1	0	0	31.37%	besar	5s
2	3	0	0	31.37%	besar	3s
3	3	0.01	0.8	31.37%	sangat besar	>10s
4	1	0.01	0.8	31.37%	sangat besar	>10s
5	1.3	0	0.3	31.37%	besar	4s
6	5	0	0.08	31.37%	kecil	3s
7	4	0	0.1	29.41%	sangat kecil	2s
8	3.5	0	0.1	29.41%	sangat kecil	1s

Tabel 4.5 menunjukkan pengujian nilai K_p , K_i , K_d dengan *tuning* manual untuk motor penggerak vertikal. Hasilnya adalah, nilai K_p , K_i , K_d terbaik adalah masing-masing 3.5, 0, 0.1. Dengan demikian kontroler terbaik untuk motor jenis yang dipakai pada Tugas Akhir ini adalah kontroler PD. Selain itu, nilai duty cycle maksimum dibatasi 19.60%. Dengan nilai-nilai ini, ketika motor mencapai *setpoint*, motor tidak bergetar.

4.3.1 Pergerakan Horisontal



Gambar 4.5 AUV Bergerak Maju Mempertahankan Posisi Horisontal

Tabel 4.6 Hasil Tuning PID Motor untuk Penggerak Horisontal

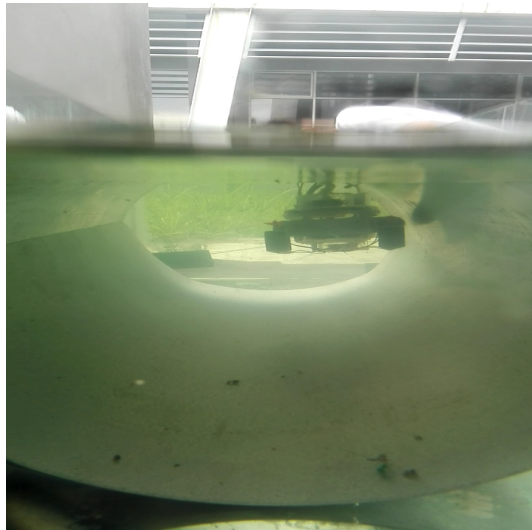
No	Kp	Ki	Kd	Duty Cycle Maksimum	Simpangan Osilasi	Lama Osilasi
1	1	0	0	19.61%	besar	7s
2	1.5	0	0.5	19.61%	sangat besar	>10s
3	3	0.01	0.5	19.61%	sangat besar	>10s
4	1	0	0	11.77%	besar	5s
5	2	0	0	11.77%	kecil	4s
6	2	0	0.1	11.77%	kecil	2s

Pada pengujian ini, Nilai Kp, Ki, dan Kd diubah-ubah dan dicari agar pergerakan motor AUV berosilasi seminimum mungkin ketika diberi suatu *setpoint*. Pada pengujian ini dilakukan di dalam sebuah akuarium agar pergerakan AUV terlihat lebih jelas. Gambar 4.5 merupakan peristiwa AUV bergerak ke kanan dan ke kiri untuk mempertahankan posisinya di tengah agar tidak menabrak dinding kanan dan kiri. Hasil pengujian yang didapatkan sebagai berikut:

Tabel 4.6 menunjukkan pengujian nilai Kp, Ki, Kd dengan *tuning* manual untuk motor penggerak horisontal. Hasilnya adalah, nilai Kp, Ki, Kd terbaik adalah masing-masing 3.5, 0, 0.1. Dengan demikian kontroler terbaik untuk motor jenis yang dipakai pada Tugas Akhir ini adalah kontroler PD. Selain itu, nilai duty cycle maksimum dibatasi 11.77%. Dengan nilai-nilai ini, ketika motor mencapai *setpoint*, motor tidak bergetar.

4.3.2 Pergerakan di dalam Pipa

Pada pergerakan robot AUV di dalam pipa, kontrol PD yang digunakan mengalami gangguan dikarenakan permukaan pipa yang tidak rata sehingga membuat pembacaan sensor jarak *infrared* mengalami pergeseran pembacaan. Respon kontrol PD yang berbeda diperlihatkan oleh AUV dikarenakan bentuk wadah yang berbeda. Pada gambar 4.6 terlihat AUV dapat mempertahankan pergerakan vertikal. Namun, pada pergerakan horizontal, AUV masih menabrak dinding pipa.



Gambar 4.6 AUV Melewati Pipa

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan pengujian dan analisis data, maka dapat ditarik kesimpulan AUV dapat bergerak di dalam pipa menggunakan kontrol PID untuk mengontrol pergerakan vertikal dan horisontal. Untuk pergerakan vertikal, AUV tidak menabrak dinding atas dan bawah. Namun, pada pergerakan horisontal, AUV masih terkendala menabrak dinding dikarenakan permukaan pipa yang tidak rata sehingga pembacaan sensor jarak mengalami pergeseran pembacaan.

Pada sensor jarak Sharp GP2Y0A41SKOF, pengukuran sensor jarak dalam air dengan persamaan karakteristik baru perlu dilakukan karena perbedaan kerapatan optik antara medium udara dan air. Dari pengujian tersebut menghasilkan error sebesar 1.25%.

Untuk AUV bergerak secara vertikal, diperlukan K_p , K_i , K_d terbaik, masing-masing sebesar 3.5, 0, 0.1. Sedangkan untuk AUV bergerak horisontal, diperlukan K_p , K_i , K_d terbaik, masing-masing sebesar 2, 0, 0.1.

5.2 Saran

Beberapa saran yang penulis dapat berikan untuk pengembangan tugas akhir adalah mekanik yang dipakai untuk *body* AUV harus lebih lebar dengan baling-baling yang lebih besar, sehingga lebih mudah untuk membuat AUV bergerak. Bisa dicoba dengan sensor jarak yang lain atau bahkan menggunakan kamera dengan resolusi tinggi sehingga lebih akurat dalam mendeteksi adanya halangan. Dapat ditambahkan sensor *gyroscope* agar AUV menjadi lebih seimbang.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Suminar, Cucum. PDAM: Tekan Kebocoran Air, Tekan Kerugian. 12 September 2014. http://www.kompasiana.com/cucum-suminar/pdam-tekan-kebocoran-air-tekan-kerugian_54f98115a3331148548b48c8.
- [2] Fela, Randy Frans. Studi Potensi Emisi Akustik Untuk Mendeteksi Kebocoran pada Pipa Distribusi Air. 2015. Yogyakarta : Universitas Gadjah Mada.
- [3] Laksana, Marttriadhi. **Micro-buble Generator**. 2008. Universitas Indonesia.
- [4] Drtil, Michael. 2006. **Electronics and Sensor Design of An Autonomous Underwater Vehicle**. The Universit of Western Australia
- [5] Koutroulis, E., dkk., “High-Frequency Pulse Width Modulation Implementation using FPGA and CPLD ICs”, Journal of Systems Architecture 52, pp. 332-344, 2006.
- [6] Kristianto, Eko. 2013. Monitoring Suhu Jarak Jauh Generator AC Berbasis Mikrokontroler. Yogyakarta : Universitas Negeri Yogyakarta.
- [7] “Controlling DC Brush Motors with H-Bridge Driver ICs”, ROHM Semiconductor, 2009.
- [8] Mulyanto, Agus, Rikendry. 2007. Sistem Kontrol Pergerakan Robot Beroda Pemadam Api. Lampung : STMIK Teknorat Lampung.
- [9] Abdullah, Salim, dkk. Rancang Bangun Kestabilan Laju Robot Kapal Selam Berbasis Mikrokontroler. Universitas Telkom.

Halaman ini sengaja dikosongkan

LAMPIRAN

Program AUV :

```
#define SETPOINT_BAWAH 10.00
#define SETPOINT_KANAN_KIRI 0.00
void GerakAtas();
void GerakKananKiri();
float tegangan_atas, tegangan_bawah;
float jarak_bawah;

float er_ab = 0, ser_ab = 0, der_ab = 0, er0_ab = 0;

float tegangan_depan;
float tegangan_kanan, tegangan_kiri;
float jarak_depan, jarak_kanan, jarak_kiri;
float delta_jarak;

float er_kk = 0, ser_kk = 0, der_kk = 0, er0_kk=0;

// KP, KI, DAN KD Atas Bawah
float kp_ab=3.5 ;
float ki_ab=0.0 ;
float kd_ab=0.1 ;

int PID_ab;

int speed1, speed0;

// KP, KI, DAN KD Kanan Kiri
float kp_kk=2.0 ;
float ki_kk=0.0 ;
float kd_kk=0.1 ;

int PID_kk;

int speed2, speed_kanan, speed_kiri;

int dir_a = 2;
```

```

int dir_b = 4;
int dir_c = 7;
int dir_d = 8;
int pwm_a = 3;
int pwm_b = 9;
int pwm_c = 10;
int pwm_d = 11;

#define FORWARD 1
#define BACKWARD 2
#define BRAKE 3
#define RELEASE 4

void setup() {
  // put your setup code here, to run once:
  speed1 = 75;
  speed0 = 75;
  speed2=30;
  speed_kanan=0;
  speed_kiri=0;

  Serial.begin(9600);
  pinMode(A1, INPUT);
  pinMode(A2, INPUT);//Sensor Bawah
  pinMode(A3, INPUT);//Sensor Kanan
  pinMode(A4, INPUT);//Sensor Kiri
  pinMode(A5, INPUT);//Sensor Depan

  pinMode(pwm_a, OUTPUT);
  pinMode(pwm_b, OUTPUT);
  pinMode(pwm_c, OUTPUT);
  pinMode(pwm_d, OUTPUT);
  pinMode(dir_a, OUTPUT);
  pinMode(dir_b, OUTPUT);
  pinMode(dir_c, OUTPUT);
  pinMode(dir_d, OUTPUT);

  Serial.println("TA Yafet");
}

```

```

Serial.println("Pengujian Gerak Atas dan Bawah");
}

void loop() {
    tegangan_bawah = (float)analogRead(A2)*5/1023;
    if(tegangan_bawah < 0.1) tegangan_bawah = 0.1;

    jarak_bawah = (22.269*tegangan_bawah*tegangan_bawah)-
    (79.479*tegangan_bawah)+75.769;
    if(tegangan_bawah==0.1)jarak_bawah=25;

    tegangan_depan = (float) analogRead(A5)*5/1023;
    if(tegangan_depan < 0.1) tegangan_depan = 0.1;

    tegangan_kanan = (float) analogRead(A3)*5/1023;
    if(tegangan_kanan < 0.1) tegangan_kanan = 0.1;

    tegangan_kiri = (float) analogRead(A4)*5/1023;
    if(tegangan_kiri < 0.1) tegangan_kiri = 0.1;

    jarak_depan = (22.269*tegangan_depan*tegangan_depan)-
    (79.479*tegangan_depan)+75.769;
    if(tegangan_depan==0.1)jarak_depan=25;

    jarak_kanan = (22.269*tegangan_kanan*tegangan_kanan)-
    (79.479*tegangan_kanan)+75.769;
    if(tegangan_kanan==0.1)jarak_kanan=25;

    jarak_kiri = (22.269*tegangan_kiri*tegangan_kiri)-
    (79.479*tegangan_kiri)+75.769;
    if(tegangan_kiri==0.1)jarak_kiri=25;

    Serial.print("Jarak: ");
    Serial.print(jarak_bawah);//tegangan_atas);
    Serial.print("\t Tegangan bawah: ");
    Serial.println(tegangan_bawah);//tegangan_bawah);

    GerakAtas();
}

```



```

delta_jarak = jarak_kanan - jarak_kiri;

if(jarak_depan >=10.00){
    speed2=30;
}
else{
    speed2=0;
}

GerakKananKiri();

motoru(3, speed2+speed_kanan);
motoru(4, (speed2+speed_kiri)*20/30);

}

void GerakAtas(){
    er_ab = SETPOINT_BAWAH - jarak_bawah;

    PID_ab = (int)(kp_ab*er_ab + ki_ab*ser_ab + kd_ab*der_ab);
    ser_ab += er_ab;
    if(ser_ab>65535 && ser_ab<-65535)ser_ab=0;
    der_ab = er_ab-er0_ab;
    er0_ab = er_ab;

    speed1 = speed0-PID_ab;//ganti + untuk gerak dari bawah ke atas
    if(speed1>75)speed1=75;
    else if(speed1<0)speed1=70;
    speed0=speed1;

    motoru(1, speed1);
    motoru(2, speed1*55/75);
    //delay(20);
}

void GerakKananKiri(){
    er_kk = SETPOINT_KANAN_KIRI - delta_jarak;

    PID_kk = (int)(kp_kk*er_kk+ki_kk*ser_kk+kd_kk*der_kk);

```

```

    ser_kk += er_kk;
    if (ser_kk > 65535 && ser_kk < -65535) ser_kk = 0;
    der_kk = er_kk - er0_kk;
    er0_kk = er_kk;

    speed_kanan += PID_kk;
    if (speed_kanan >= 30) {speed_kanan = 30;}
    else if (speed_kiri <= -20) {speed_kiri = -20;}
    }
    speed_kiri -= PID_kk;
    if (speed_kiri >= 30) {speed_kiri = 30;}
    else if (speed_kanan <= -20) {speed_kanan = -20;}

}

void motor(int nMotor, int command, int speed)
{
    int motorA, motorB;
    int pwm;

    if (nMotor >= 1 && nMotor <= 4)
    {
        switch (nMotor)
        {
            case 1:
                motorA = dir_a;
                pwm = pwm_a;
                break;
            case 2:
                motorA = dir_b;
                pwm = pwm_b;
                break;
            case 3:
                motorA = dir_c;
                pwm = pwm_c;
                break;
            case 4:
                motorA = dir_d;
                pwm = pwm_d;

```

```

        break;
default:
    break;
}

switch (command)
{
case FORWARD:
    digitalWrite (motorA, HIGH);
    analogWrite(pwm,speed);
    break;
case BACKWARD:
    digitalWrite (motorA, LOW);
    analogWrite(pwm,speed);
    break;
case BRAKE:
    digitalWrite (motorA, LOW);
    analogWrite(pwm,0);
case RELEASE:
    digitalWrite (motorA, LOW);
    analogWrite(pwm,0);
default:
    break;
}
}
}

void motoru(int xMotor, int speedmotor)
{
    if(speedmotor >= 0 && speedmotor <= 255)
    {
        motor(xMotor, FORWARD, speedmotor);
    }
    if(speedmotor >= -255 && speedmotor <= -1)
    {
        speedmotor = map(speedmotor, -255, -1, 255, 1);
        motor(xMotor, BACKWARD, speedmotor);
    }
}
}

```

RIWAYAT HIDUP PENULIS



Yafet Yulio lahir di Surabaya pada 15 Juli 1994. Anak kedua dari 2 bersaudara dari pasangan Budi Jatmiko dan Sulistyorini. Penulis menyelesaikan pendidikan dasar di SDK Santo Yosef Surabaya, dilanjutkan dengan pendidikan menengah di SMP Kr. Petra 2 Surabaya dan SMA Kr. Petra 1 Surabaya. Pada tahun 2012, penulis memulai pendidikan di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Selama kuliah penulis aktif membantu penyelenggaraan kegiatan dan aktif sebagai asisten laboratorium Elektronika Dasar dan praktikum Elektronika pada semester ganjil dan genap 2015-2016.

Email :

yafet94@gmail.com.com

Halaman ini sengaja dikosongkan